

【特許請求の範囲】

1. サーバに結合されたディスクに記憶されたアプリケーションプログラムを実行するためにクライアントがインターネットサイトのサーバに接続することができる方法において、
前記サーバに接続するデータトランスポートインタフェースを提供するステップと、
前記サーバから前記サーバのネットワークアドレスを示すサーバアドレス信号および前記アプリケーションプログラムを含むファイルシステムを示すパス名信号を移送するステップと、
前記パス名信号の機能としての取り付け要求を発生するステップと、
前記取り付け要求を前記サーバに移送し、前記アプリケーションプログラムを含む前記ファイルシステムを示すファイル記述子を指示するサーバファイルポインタのアレイを提供するように、前記サーバに指令するステップと、
前記サーバアドレス信号、前記パス名信号の機能としての遠隔ファイルポインタのアレイおよびサーバファイルポインタの前記アレイを生成するステップと、
前記遠隔ファイルポインタのアレイを前記クライアントで作動するプログラムロード要素に供給するステップと、
前記ディスクから前記選択された遠隔ファイルポインタに関連するファイルブロックをローカルメモリ要素に移送するために前記遠隔ファイルポインタの中の1つを使用するように前記プログラムロードに指令するステップとからなることを特徴とする方法。
2. 前記クライアントによって使用されたファイルシステムプロトコルを示すクライアントタイプ信号を前記サーバに移送する他のステップを含むことを特徴とする請求項1に記載の方法。
3. 管理プログラムを前記サーバに供給し、選択ファイルシステムプロトコルにより前記ディスクにコンピュータファイルのファイルシステムを記憶する他のステップを含むことを特徴とする請求項1に記載の方法。

4. 前記サーバに管理プログラムを供給する前記ステップが、前記ファイルコントローラを提供し、DOS、UNIX、VMS、WINDOWS、WINDOWS 95、WINDOWS NT、OS/2、SYSTEM7 ファイルシステムプロトコルの中の1つあるいはそれ以上によりコンピュータファイルを記憶するステップを含むことを特徴とする請求項3に記載の方法。
5. 管理プログラムを供給する前記ステップが、ディスク分割を生成する分割コントローラを提供し、前記選択ファイルシステムプロトコルによりコンピュータファイルのファイルシステムを記憶するステップを含むことを特徴とする請求項3に記載の方法。
6. 管理プログラムを供給する前記ステップが、
コンピュータファイルの前記ファイルシステムを示すディレクトリ情報信号を発生し、かつ前記ディレクトリ情報信号を前記クライアントに移送するファイルディレクトリコントローラを提供し、前記クライアントによってアクセスされるのに利用可能なコンピュータファイルのファイルディレクトリを提供するステップを含むことを特徴とする請求項3に記載の方法。
7. 前記サーバにファイルアクセス制御プログラムを供給し、クライアント特権信号の機能として、前記クライアントによって取り付けられるのに利用可能なコンピュータファイルのファイルシステムのセットを決定する他のステップを含むことを特徴とする請求項1に記載の方法。
8. 前記クライアントが、HTTPサーバに結合されたメモリ装置に記憶されたアプリケーションプログラムにアクセスし、前記ファイルシステム要求をネットワークにわたって伝送するためにHTTPコンプライアント信号に変換するリダイレクタ要素を提供する他のステップを含むことを特徴とする請求項1に記載の方法。
9. ファイルシステム要求に応じて、URLパス名を発生するように前記リダイレクタ要素を指令し、遠隔ファイルシステム内部に記憶されたファイルにアクセスする信号を生成するステップを含むことを特徴とする請求項8に記載の方法。
10. 複数のリダイレクタ要素を提供し、前記ファイルシステム要求を選択された

トランスポートプロトコルに変換する他のステップを含むことを特徴とする請求項 8 に記載の方法。

11. 複数のリダイレクタ要素を提供する前記ステップが、NFS リダイレクタ要素を提供し、要求を NFS コンプライアント信号に変換するステップを含むことを特徴とする請求項 10 に記載の方法。
12. 複数のリダイレクタ要素を提供する前記ステップが、SMB リダイレクタ要素を提供し、要求を SMB コンプライアント信号に変換するステップを含むことを特徴とする請求項 10 に記載の方法。
13. 前記複数のリダイレクタ要素の中の 1 つを選択し、前記ファイルシステム要求を変換する他のステップを含むことを特徴とする請求項 10 に記載の方法。
14. 1 つあるいはそれ以上のネットワークのサーバで作動する遠隔ファイルシステムのタイプの機能あるいは前記クライアントに結合された 1 つあるいはそれ以上のネットワークによって使用されるネットワークトランスポートプロトコルタイプの機能として、前記リダイレクタを選択するステップを含むことを特徴とする請求項 13 に記載の方法。
15. 前記クライアントが複数のネットワークに結合し、かつ前記複数のネットワークの各々に対する伝送待ち時間特性を比較する機能として前記リダイレクタを選択する他のステップを含むことを特徴とする請求項 13 に記載の方法。
16. 前記クライアントに前記アプリケーションプログラムを含む遠隔ファイルシステムの一部を記憶するキャッシュのためのキャッシュメモリへのアクセスを提供し、かつキャッシュリダイレクタ要素を提供し、前記キャッシュメモリ内部に記憶された前記ファイルシステムの前記一部にアクセスするように前記クライアントに指令する他のステップを含むことを特徴とする請求項 1 に記載の方法。
17. 前記ファイルシステムに反映するように前記キャッシュメモリに指令するステップを含むことを特徴とする請求項 16 に記載の方法。
18. 前記キャッシュメモリ内部に記憶されたアプリケーションプログラムの少なくとも 1 つのリンクを有するオフラインページを提供するステップを含むことを特徴とする請求項 16 に記載の方法。

19. 前記クライアントが前記キャッシュメモリ内部にアプリケーションプログラムを選択的に記憶することができるプリロード制御装置を提供する他のステップを含むことを特徴とする請求項16に記載の方法。
20. 前記サーバのプログラムメモリ内部の前記アプリケーションプログラムを実行するように前記サーバに指令する処理を提供するステップを含むことを特徴とする請求項1に記載の方法。
21. 前記アプリケーションプログラムに関連したファイルを示す情報および前記アプリケーションプログラムの前記ファイルに関連したファイル構造を示す情報および／または前記アプリケーションプログラムのための動作環境を示す情報を有するアプリケーション情報ファイルを生成するステップを含むことを特徴とする請求項1に記載の方法。
22. 前記サーバに第2のサーバのリンクを提供する他のステップ、および前記第2のサーバに記憶され、かつ前記クライアントによる実行に利用可能な前記クライアントアプリケーションプログラムを識別するように前記サーバに指令する他のステップを含むことを特徴とする請求項1に記載の方法。
23. 前記サーバが前記アプリケーションプログラムへの前記クライアントのアクセスを制御できるアクセス制御処理を提供する他のステップを含むことを特徴とする請求項1に記載の方法。
24. インターネットサイトのサーバに結合されたディスクに記憶されたアプリケーションプログラムをクライアントが実行するシステムにおいて、
前記サーバに接続するデータトランスポートインタフェースと、
前記サーバから前記サーバのネットワークアドレスを示すサーバアドレス信号および前記アプリケーションプログラムを含むファイルシステムを示すパス名信号を収集する接続手段と、
前記パス名信号の機能としての取り付け要求を発生し、かつ前記アプリケーションプログラムを含む前記ファイルシステムを示すファイル記述子を指示するサーバファイルポインタのアレイを提供するように前記サーバに指令するために前記取り付け要求を前記データトランスポート手段に供給する取り付け手段と、

前記サーバアドレス信号、前記パス名信号の機能としての遠隔ファイルポインタのアレイおよび前記サーバファイルポインタのアレイを生成するファイル操作手段と、

前記遠隔ファイルポインタの中の1つを選択し、かつ前記選択された遠隔ファイルポインタに関連するファイルブロックを前記ディスクからローカルプログラムメモリ要素に移送する手段を有するプログラムロード要素とを備えていることを特徴とするシステム。

25. 前記ファイル操作手段が、前記サーバアドレス、前記パス名信号および前記ファイルポインタの1つを連結し、前記遠隔ファイルポインタの1つを生成する連結手段を含むことを特徴とする請求項24に記載のシステム。

26. 前記アプリケーションプログラムが第2のアプリケーションプログラムをインストールする導入プログラムを含み、さらに、

前記導入プログラムに関連したファイルを示すアプリケーション情報を生成する処理と、

前記アプリケーションプログラムをインストールするために前記クライアントからの要求を検出する検出器と、

前記アプリケーション情報ファイルに応じて前記クライアントの動作環境を構成し、それによって前記第2のアプリケーションプログラムが前記クライアントで実行する前記導入プログラムの動作によって前記クライアントにインストールされる構成要素とを含んでいることを特徴とする請求項24に記載のシステム。

【発明の詳細な説明】

サーバに結合されているメモリ装置に格納された アプリケーションプログラムを実行するシステムおよび方法

発明の分野

本発明は、コンピュータをコンピュータネットワークに接続するシステムおよび方法、特に、クライアントがサーバサイトに記憶されたアプリケーションプログラムを実行するシステム

および方法に関するものである。

発明の背景

ディジタルコンピュータは、ユーザがどのような公知の技術によっても実現できない容量および速度でデータを編成し、記憶し、解析することができる高性能のデータ処理ツールである。コンピュータネットワークは同様な先進手段であった。いくつかのコンピュータと一緒に結合し、共有リソースおよびクロスプラットフォームを提供することによって、エンジニアは、コンピュータネットワークに十分な処理能力を与え、会社および大学全体を支援するツールに発展させた。個別のマイクロコンピュータに関するコンピュータネットワークの性能の増加は、分散処理によってもたらされる。分散処理は、コンピュータネットワークに接続された異なるワークステーション間のコンピュータ作業負荷の分割あるいは分散を可能にする。

最も幅広く容認され、おおいに使用されているコンピュータネットワークはインターネットである。インターネットは、単一の世界的なネットワークに形成された相互接続コンピュータネットワークのグローバルシステムである。ユーザは、インターネットを通して異なる国々のユーザに対してメッセージを対話式で送信できる。同様に、米国のユーザは、ヨーロッパおよびアジアのライブラリに格納されたファイルにアクセスでき、個人使用のためにこれらのファイルをダウンロードすることができる。したがって、インターネットは、ハム無線技術者によ

つ

てもたらされる通信機能と同様な強力な通信機能を提供する。さらに、インター

ネットは、共通のライブラリと同様に作動し、世界中のインターネットサイトから使用可能なリソースおよび情報への電子的アクセスをもたらす。

インターネットは、通信およびファイル共有のための優れた発展したリソースを提供するけれども、インターネットはなお基本的には静的媒体である。換言するに、インターネットのコンテンツ、すなわちインターネット上のサイトをアクセスするユーザに使用可能なリソースは多くは記憶されたデータである。一般的には、ユーザは、Webブラウザプログラムを使用してインターネットの周りを巡回する。Webブラウザプログラムは、テキスト画像、サウンドクリップおよびビデオクリップを含む記憶データのマルチメディア表示を可能にする。これによって、ユーザは、インターネット上の異なるサイトにインターネットを介して接続できる。しかしながら、ユーザはインターネットサイトに記憶された情報を見て、それ自身の個人使用のためにコピーを作成できるけれども、現在、ユーザはアプリケーションプログラムを直接実行するのに適しているシステムを欠いている。それ自体、インターネットは、一般に分散処理の長所をもたらすことができない。

インターネットのクライアント/サーバアプリケーションの可用性を妨害する要因のひとつに、インターネットのサイズが挙げられる。特に、インターネットは非常に膨大であり、開放されているために、会社および大学は、その内部サーバとインターネットとの間にファイアウォールを配置した。これらのファイアウォールは、このようなNFSおよびSMBのような現在のシステムがアプリケーションにアクセスすることを防止することによって有用なアプリケーションへのアクセスを抑止する。それ自体、ファイアウォールは、従来のネットワークシステムの使用がクライアント/サーバアプリケーションを実行することを防止するバリアの役目を果たす。

この問題を解決するために、アプリケーションプログラムを記憶するインターネットサイトとアプリケーションプログラムを使用したいユーザにサービスするインターネットサイトとの間をアプリケーションプログラムが移動するシステムが提案された。その1つのシステムにサンマイクロシステムズによって提案され

たJ A V Aシステムがある。J A V Aシステムは、遠隔インターネットサイトに置かれたアプレットと呼ばれる小さいアプリケーションプログラムをダウンロードし、実行するためにインターネットユーザによって操作することができる解釈言語（インタプリタ）システムを提供する。したがって、J A V Aシステムは、インターネットサイトに置かれたアプリケーションプログラムがインターネットユーザに配分するために利用される点で分散処理の形式を提供する。これは、記憶データを単に参照して、転送することしか許されなかった、従来のインターネットユーザにしてみれば、著しく進歩した機能である。

しかしながら、J A V Aは従来技術の改良を提供するけれども、J A V Aシステムはいくつかの顕著な欠点を持っている。まず第一に、J A V Aシステムは、インターネットユーザにダウンロードされ、インターネットサイトで実行されるべき各アプリケーションがJ A V Aプログラミング言語で記述されなければならないことである。したがって、マイクロソフトWord、マイクロソフトExcel、Intuit Quicken、あるいは他の開発されたアプリケーションプログラムのような使用可能なプログラムアプリケーションは、J A V Aシステムと併用するに当たってJ A V A互換プログラミング言語で再記述されねばならない。これは、もちろん、インターネットにわたるアプリケーションプログラムの配置に対する直接の障害を生じる。さらに、これらの再記述されたアプリケーションが検証済みのコードでなく、プログラムの再記述中に作りこまれたバグあるいはJ A V Aシステムに潜むバグのどちらかにより必然的に失敗する場合、かなりのデバッグの努力が必要である。さらに、提案されたJ A V Aシステムは解釈言語システムであり、解釈言語システムは周知の通りに遅い。したがって、インターネットユーザは、これらのアプリケーションを実行している間、かなりの動作速度の減少に気づく可能性がある。さらに、インターネットサイトに配置されるJ A V Aアプリケーションは、特別にインターネット配置用として作成される。したがって、これらのJ A V Aアプリケーションプログラムは、一般的にはユーザによって使用されるプログラムではない。その代わりに、インターネットにわたって配置されたこれらのJ A V Aプログラムとは異なる非J A V Aプログラムをユーザは操作する可能性がある。したがって、遠隔サイトのインターネ

ットユーザとホストサイトのユーザとの間に互換性の亀裂が生じる。それ自体、J A V Aシステムは、遠隔地のユーザとホストサイトのユーザとの間に障壁を本来生じる。

したがって、本発明の目的は、インターネット上におけるアプリケーションプログラムの配置および実行を可能にするシステムを提供することにある。

本発明の目的は、インターネットファイアウォールを通り抜けることができるシステムを提供することにある。

本発明の目的は、遠隔ユーザをホストサイトのインターネットネットワークに結合するシステムに提供することにある。

本発明のさらにもう一つの目的は、アプリケーションを再記述することなく、インターネット上に既存のアプリケーションプログラムを配置するシステムを提供することにある。

本発明の他の目的は、遠隔サイトで迅速に実行するプログラムを提供するインターネット上にアプリケーションを配置するシステムを提供することにある。

本発明のこれらおよび他の目的は本発明の下記の説明から明らかにされる。

発明の概要

本発明によるシステムおよび方法によって、一般的にクライアントと呼ばれるコンピュータは、インターネットサイトのサーバに接続して、サーバに結合されたメモリ装置に記憶されているアプリケーションプログラムを実行することができる。

本発明の方法は、クライアントがH T T Pサーバに結合されたメモリ装置に記憶されたアプリケーションプログラムを実行できる方法を含んでもよく、アプリケーションプログラムを含み、かつH T T Pサーバに結合された遠隔ファイルシステムを備えるステップと、オペレーティングシステムを監視し、遠隔ファイルシステム内に記憶されたファイルに対するファイルシステム要求を検出するステップと、ファイルシステム要求をネットワークにわたって伝送するH T T Pコンプライアント信号に変換するリダイレクタ要素に、ファイルシステム要求を渡すステップとを含んでもよい。本発明のこの実施において、取り付けるステップは、

アプリケーションプログラムに関連するファイルのポインタを示すファイルポインタの配列を生成するステップを含んでもよい。

他の実施において、本発明の方法は、ファイルシステム要求を複数のトランスポートプロトコルの中の一つに選択されたトランスポートプロトコルに変換する複数のリダイレクタ要素を提供するステップを含んでもよい。例えば、クライアント要素は、その各々がNCP、NFS、SMBあるいはネットワークメディアにわたってファイルのようなデータを伝送あるいは共有するプロトコルを提供するオープンサービスあるいは独自サービスのいずれかの、異なるファイルトランスポートプロトコルを使用する複数の異なるネットワークのネットワークインタフェースを有してもよい。この実施において、各々のリダイレクタ要素は、データトランスポートプロトコルの一つあるいはそれ以上と併用するのに適合されてもよく、それによって、クライアント要素は、各々が異なる遠隔ファイルシステムを有する複数のサーバのいずれか一つにアクセスできる。この方法は、ファイルシステム要求を変換する複数のリダイレクタ要素の一つを選択するステップを含んでもよい。リダイレクタ要素を選択するステップは、ネットワークの伝送プロトコルと同様にネットワークの各々の伝送待ち時間の基準としても、またファイルシステムのタイプを含む任意の適当な基準あるいは特性の目的として行われてもよい。本発明の一つの実施において、複数のリダイレクタには、全てのリダイレクタ要素にはクライアントコンピュータのオペレーティングシステムの同じインタフェースが装備されるようにリダイレクタ要素の各々をある程度までカプセルに入れるラッパインタフェースが装備されている。これは、移植性をもたらし、全システムの複雑性を減らすことがわかる。

ここで使用されるような用語“リダイレクタ”は、データあるいはサービスに対する要求を選択されたデータ源あるいはサービス源に向けることができ、ファイルシステム要求をネットワークメディアにわたるデータの伝送のための少なくとも一つのネットワークプロトコルに従うコマンドあるいはデータ信号に任意に変換できるプログラムあるいは処理を含むことが分かる。

他の実施例において、本発明の方法は、アプリケーションプログラムを含むファイルシステムの一部を記憶するキャッシュのためのキャッシュメモリへのアク

セスをユーザにもたらす他のステップと、キャッシュメモリ内部に記憶されたファイルシステムの一部をアクセスすることによってクライアントファイル要求に応答するキャッシュリダイレクト要素を供給する他のステップとを含んでいる。ファイルシステムのアクセス部分は、クライアントローカルプログラムメモリ内部のアプリケーションプログラムを実行するために使用されてもよい。1つの実施において、この方法は、それに記憶されているファイルシステムに反映するようにキャッシュメモリを向けるステップを含んでもよい。したがって、この実施において、ミラーリング処理は、変化がサーバによって送り出されたファイルシステム内部で検出されるとき、ファイル、およびキャッシュメモリ装置内部に保持されるファイルあるいはディレクトリ構造が同様に更新されるように作動できる。オプションとして、このミラーリングは、自動的あるいはユーザの命令によって行うことができる。

他の実施において、本発明の方法は、サーバによって使用されるファイルあるいはディレクトリ構造によるファイルシステムの一部を記憶するステップを含んでもよい。したがって、キャッシュメモリ装置は、これらのファイルに関連するディレクトリ構造を示す情報だけでなく、アプリケーションプログラムを実行するファイルを示す情報を記憶できる。このディレクトリ構造は、ディレクトリ、サブディレクトリ、パスおよび他のディレクトリ構造を含んでもよい。さらに、キャッシュのディレクトリ構造はサーバによって使用されるファイルシステムに適応できるので、例えば、ウィンドウズクライアントはキャッシュメモリにUNIXファイル構造を記憶できる。

他の実施において、本発明の方法は、キャッシュメモリ内部に記憶されているアプリケーションプログラムに対する少なくとも1つのリンクを示しているオフラインページを提供するステップを含んでもよい。このオフラインページは、クライアントによってアクセス可能なメモリ装置内部に記憶され、かつクライアントのキャッシュメモリ装置内部に記憶されたファイルに接続するのに利用されるHTMLページであってもよい。オフラインページの呼び出しの際に、ユーザは、キャッシュメモリ装置内部に記憶されたアプリケーションプログラムの実行をさせるようにページ内部のリンクを起動できる。本発明の他の実施において、本

明の方法は、クライアントがキャッシュメモリ内部にアプリケーションプログラムを選択的に記憶できるプリロード制御を提供するステップを含んでもよい。したがって、このプリロード制御によって、クライアントは、サーバサイトに記憶され、かつクライアントに送り出すためにサーバによって利用され、コンピュータネットワークを通してアクセスする必要なしにクライアントによって使用するためにキャッシュメモリ装置内部に記憶されているアプリケーションプログラムを選択できる。

他の実施において、この方法は、サーバのプログラムメモリ内部のアプリケーションプログラムを実行するようにサーバを向ける方法を提供するステップを含んでもよい。このように、本発明の方法は、プログラムがサーバプログラムメモリ内部で実行している間、クライアントシステムが端末の役目を果たすことができ、それによってユーザがクライアントプログラムメモリ内部のアプリケーションプログラムを実行するかあるいはその代わりにサーバプログラムメモリ内部のアプリケーションプログラムを実行するかの間で選択できる機能性も含んでもよい。

他の実施例において、本発明は、サーバが、サーバに結合されたメモリ装置に記憶されたアプリケーションプログラムの遠隔実施をもたらすことができる方法として理解できる。この方法は、アプリケーションプログラムに関連するファイルを示すアプリケーション情報ファイルを生成するステップを含んでもよい。他のステップにおいて、この方法は、アプリケーション情報ファイルのポインタを示すリンクを発生し、アプリケーションプログラムを実行するようにクライアントからの要求を検出する。この要求に応じて、本発明の方法は、アプリケーションプログラムに関連するファイルを示すデータ信号をクライアントに伝送できる。本発明の1つの実施において、アプリケーション情報を生成するステップは、アプリケーションプログラムのファイルに関連するファイルあるいはディレクトリ構造を示す情報を生成するステップを含んでもよい。このように、アプリケーション情報ファイルは、実行するのに適しているフォーマットでアプリケーション

ンプログラムを保持するサーバによって使用されるファイルあるいはディレクトリ構造だけでなく、アプリケーションプログラムに関連するファイルを示す情報を

含んでもよい。

本発明の方法は、アプリケーションプログラムのためのオペレーティング環境を示す情報を含むアプリケーションファイルを生成するステップも含んでもよい。このように、アプリケーション情報ファイルは、クライアントがアプリケーションプログラムに適応させるようにローカルオペレーティング環境を再構成できる情報を含んでもよい。

他の実施例において、本発明の方法は、サーバに第2のサーバへのリンクを提供するステップと、第2のサーバに記憶され、クライアントによる実行に利用されるこれらのアプリケーションプログラムをクライアントのために識別するようにサーバに指令するステップとを含んでもよい。これによって、サーバは、他のサーバによって利用され、かつクライアントによって呼び出すことができるこれらのアプリケーションプログラムをクライアントに知らせることができる。これは、クライアントに複数の異なるサーバで利用可能なアプリケーションへのアクセスを提供する。任意には、サーバは、他のサーバで利用可能なアプリケーションプログラムに対するリンクを形成できる。クライアントは、これらのリンクを使用し、これらのアプリケーションプログラムを実行する。

他の実施例において、本発明の方法は、アプリケーションプログラムを実行するクライアントの数を監視する計測方法を提供するステップも含んでもよい。任意には、計測方法は、アプリケーションの使用の所定の特性を識別し、かつリソースをより適切に割り当てるためのような管理機能のためのシステム管理者に提供できる統計情報を生成できる。

さらにもう1つの実施例において、本発明の方法は、サーバがアプリケーションプログラムへのクライアントのアクセスを制御できるアクセス制御方法を提供するステップも含んでもよい。本実施例において、サーバは、このサーバあるいは他のサーバによって保持されるアプリケーションプログラムにアクセスできる

これらのクライアントを選択されたグループに制限するためにパスワードあるいは他のキーのいずれかによって認証を必要とするアクセス制御処理を含んでもよい。

他の実施例において、本発明の方法は、クライアントがサーバに結合されたメ

モリ装置に記憶されたアプリケーションプログラムをインストールできる処理を含んでもよい。これらの方法は、サーバによる操作のためのアプリケーションプログラムをインストールする導入プログラムへのアクセスをサーバに提供するステップと、導入プログラムに関連したファイルを示すアプリケーション情報ファイルを生成するステップと、アプリケーションプログラムを局部的にインストールするためにクライアントからの要求を検出するステップと、アプリケーション情報ファイルに応じて、導入プログラムに関連したファイルを示すデータをクライアントに伝送するステップとを含んでもよい。本発明のこの実施において、この方法はアプリケーションプログラムに関連したファイルを示す情報を有するアプリケーション情報ファイルを生成するステップも含んでもよい。

他の実施例において、本発明は、クライアントがサーバのメモリ装置に記憶されたアプリケーションプログラムを局部的に実行できるサーバ処理として理解できる。これらの方法は、アプリケーションプログラムに関連したファイルを記憶し、アプリケーションプログラムの実行可能なコードを含む1つあるいはそれ以上のサーバディレクトリのための記憶域を有するメモリ装置にアクセスできるHTTPサーバプロセスを提供するステップを含んでもよい。サーバは、クライアントに、ファイルおよびディレクトリへのアクセスを提供し、クライアントは、アプリケーションプログラムを含むファイルシステムの一部をキャッシュし、キャッシュメモリ内部に記憶されているファイルシステムの一部にアクセスするためのキャッシュリダイレクタ要素を使用できる。

ここで使用されるような用語“クライアント”は、本発明による方法を作動するのに適しているいかなるデータ処理システムも含んでいる。用語“サーバ”は、コンピュータネットワークに接続し、HTTP、FTPの下であろうと任意の他の従来のプロトコルあるいは所有権の主張できるプロトコルの下であろうと、

アクセス要求を受け取るデータ処理プラットフォームで実行するいかなる処理も含んでもよい。ここで使用されるような用語“アプリケーションプログラム”は、コンピュータシステムの処理装置によってアクセスされ、かつ処理されるためのフォーマットのデータを含む任意のコンピュータファイルあるいはファイルのセットを含んでいる。ここで使用されるような用語“メモリ装置”は、コンピュータ

の読み出し可能なデータを記憶でき、記憶データをアクセスするインタフェースを提供する任意の装置を含み、かつ揮発性メモリ装置および固定記憶装置を含み、ハードディスク、RAM、ネットワークディスク、および任意の他の適当なデータ記憶システムを含むことがわかる。

ここで使用されるような用語“ファイルシステム”は、コンピュータシステムに記憶されるかあるいはコンピュータシステムに結合されているファイル、ディレクトリ、ディスク、装置およびポート間の関係を規定する論理構造を含み、かつ記憶ファイル、ディスク、ディスク分割、ファイルブロック、コンピュータデータの装置あるいは他のソース間の階層関係を規定するディレクトリ構造のファイルシステムを含んでいる。

用語がここで使用されるような用語“サーバアドレス信号”は、サーバのネットワークアドレスを識別するいかなる記述も含んでいる。用語は、任意のホスト番号、任意のホスト名、任意のドメイン名あるいはサーバのネットワークアドレスを識別するのに十分な任意の他の記述を含むサーバのインターネットアドレスに対する任意の記述を含むことがわかる。ここで使用されるような用語“パス名信号”は、ファイル、ディレクトリ、サブディレクトリ、装置、ポートあるいは任意の他のディレクトリプリミティブを含むファイルシステムの1つあるいはそれ以上の要素を識別する情報をオペレーティングシステムに提供するのに適している任意の記述を含んでいる。

本発明の他の実施例において、システムは、選択ファイルシステムプロトコルにより記憶されたコンピュータファイルの少なくとも1つのファイルシステムをディスクに記憶する管理プログラムを有するサーバ要素を含んでもよい。1つの

このような管理プログラムは、DOS、UNIX、VMS、WINDOWS、WINDOWS 95、WINDOWS NT、OS/2、あるいはSYSTEM 7 ファイルシステムプロトコルの1つあるいはそれ以上によるファイルシステムのコンピュータファイルを記憶する手段を含んでもよい。本発明の他の実施例では、このシステムは、サーバのHTTP処理のポートの呼び出しを生成する要素を含む接続要素を含んでもよい。例えば、この接続要素は、サーバ処理のポート80の呼び出しを生成する接続プログラムを含んでもよい。ポート80、すなわち周

知のポートはクライアントシステムに結合されたプロセスを操作するHTTPサーバのインターネットアドレスを供給する。

本発明の他の実施例において、サーバはファイル共有プログラムを含んでいる。ファイル共有プログラムは、NFS、SMBあるいはHTTPコンプライアントであってもよく、ファイル共有プログラムは、サーバシステムに記憶されているコンピュータファイルのポインタのアレイとして理解できるUnix inodeのようなファイルポインタのアレイを作成できる。

本発明のこれらの実施例および他の実施例は、下記の詳細な説明および同じ参照番号は同じ要素を示す添付図面を参照することによってより明らかに理解される。

図面の簡単な説明

図1は、インターネットサイトに接続を生じさせ、このサイトに置かれたアプリケーションプログラムを実行する本発明による1つのシステムを示している。

図2は、HTTPサーバでアプリケーションプログラムのリンクを形成する1つの処理を示している。

図3は、図1に示されたシステムをより非常に詳細に示している。

図4は、備えられた論理ディスク分割を有する図1のシステムを示している。

図5は、インターネットサイトに記憶されたアプリケーションプログラムのエントリポイントをクライアントプログラムローダに提供するクライアントファイルシステムの階層に統合するファイルポインタのアレイを有する図1のシステム

を示している。

図6は、本発明によるシステムの他の実施例を示している。

図7は、アプリケーションプログラムにアクセスするキャッシュメモリを使用する図6のシステムを示している。

図8は、アプリケーションプログラムにアクセスし、実行するディレクトリ構造を提供する図1および図6のシステムと併用するのに適しているグラフィックインタフェースを示している。

図9は、Webインターネットサイトからのアプリケーションプログラムを実

行する本発明による方法の機能ブロック図である。

発明の詳細な説明

本発明は、なにかんずく、Webサイト管理者がそのWebページ内部の遠隔アプリケーションにリンクを提供できるシステムおよび方法を備えている。本発明の1つの実現は、遠隔クライアントがHTTPサーバに記憶されたアプリケーションプログラムを実行することができることである。この目的のために、本発明は、サーバに遠隔ファイルシステムへのアクセスおよびアプリケーションプログラムに関連したファイル、ディレクトリ構造およびシステム環境の情報を含み得るファイルへのアクセスを提供する。本発明は、クライアントがアプリケーションプログラムを含む共有ファイルシステムの一部を備えることができるシステムも含んでもよい。このシステムは、共有ファイルシステムの一部にアクセスするオペレーティングシステム要求をHTTPコンプライアントコマンドに変換するHTTPリダイレクタ要素を含んでもよい。インターネットサイトのHTTPサーバは、コマンドに応答し、HTTPサーバディレクトリへのアクセスを提供する。したがって、HTTPリダイレクタは、サーバディレクトリへの駆動接続を作成し、クライアントにそれに記憶されたファイルへのアクセスを提供する。これはHTTP共有ファイルシステムを提供する。このように、遠隔アプリケーションは、クライアントファイルシステムの中に取り付け、それから、ブラウザプログラムのウィンドウ内部あるいは開始プログラムから実行できる。任意には、リンクは、ブラウザあるいはランチャプログラムのウィンドウ内部のプログラム

アイコンとして表示できる。

本発明によるシステムおよび方法の1つの長所は、会社のネットワーク内部で実行するのに利用されるレガシーアプリケーションがインターネット、イントラネット、あるいは他のネットワークシステムにわたって容易に開始し、ブラウザあるいはランチャプログラムのウィンドウ内部で実行できることであることが分かる。

この目的のために、本発明のシステムおよび方法は、ブラウザタイププログラム内部に表示され、かつネットワーク接続のサーバ側に保持され、アプリケーション

ョンプログラムを実行する情報を含むアプリケーション情報ファイルにクライアントを結合できるリンクを形成できる能力を提供する。アプリケーション情報ファイルは、クライアントがクライアントのプログラムメモリ内部のアプリケーションを実行するために使用する*.exeファイルおよび*.dllファイルのようなこれらのファイルを識別できる。1つの実施において、サーバは、共有ファイルシステムを介してクライアントが使用可能なこれらのファイルを作成する。任意には、クライアントは、共有ファイルシステムからファイルをダウンロードし、クライアントに接続されているキャッシュメモリシステム内部にファイルを記憶するために転送プロトコルを使用できる。クライアントは、局所的にキャッシュメモリからのアプリケーションを任意に実行できる。それとは別に、クライアントで操作するプログラムローダ処理は、クライアントのローカルメモリにアプリケーションプログラムの実行可能なコードのブロックをロードでき、このコードはこのローカルメモリから実行できる。

次に、本発明は、典型的であり、本発明の限定あるいは完全な表現として理解されるべきでないある例示的な実施例を参照して説明される。

図1は、サーバ、特にHTTPコンプラアントサーバに結合された駆動装置に記憶されたアプリケーションプログラムをアクセスし、配置し、実行する本発明によるシステム10を示している。特に、Web管理者は、インターネットサイトおよびイントラネットサイトを含む異なるネットワークロケーションに記憶されたアプリケーションプログラムへのアクセスをユーザに提供するために本発明

のシステムを使用できる。図1に示されるように、このシステム10は、クライアント要素12と、コンピュータLAN13aおよび13bと、HTTPサーバ14と、メモリ装置14aおよび14bと、サーバ15と、ネットワークノード17と、オプションプログラムインタフェース16と、オプションアプリケーションアイコン16a、16bおよび16cと、通信要素18とネットワークノード19aおよび19bとを含んでいる。

図1に示されたクライアント要素12は、プロセスを実行し、通信リンクにサーバサイトを提供する通信インタフェースを作動させるのに適している市販されているPCコンパチコンピュータ、Unixワークステーションあるいは他のデ

ータ処理システムであってもよい。さらに、クライアント12は、スタンドアロンコンピュータシステムあるいは、図示されるように、ネットワークコンピュータシステムであってもよい。クライアント12は、モデムのような電気通信要素18を含んでもよいし、HTTPサーバとの接続を確立するのに適しているネットスケープブラウザのようなブラウザ、あるいはランチャ、プログラムも含んでもよい。図1に示された実施例において、ブラウザは、ユーザが選択ファイルにアクセスし、実行するシステム10の動作を制御し、命令できるように、グラフィカルユーザインターフェイス16を、オプションとして提供する。図示されたグラフィックインタフェース16は、ユーザがサーバ14に結合されたディスク14aのようなメモリ装置に記憶された関連アプリケーションプログラムを起動するグラフィック装置を提供する16aのようなアプリケーションアイコン要素を表示する。

図1は、通信要素18およびメモリ装置14aおよび14bに接続するサーバ要素14を含むHTTPサーバサイトを示している。サーバ14は、クライアント、例えばクライアント14によってアクセスできるHTTPサーバ処理を実行するのに適している任意のサーバ要素であってもよい。一般的には、サーバ14は、ネットワークアクセスのための1つあるいはそれ以上のファイルを含んでいるインターネットサイトあるいはイントラネットサイトを保有する1つあるいはそれ以上のサーバ処理を実行するデータ処理システムである。図1に示された実

施例において、サーバ14は、外部ディスク駆動システムとして示されている2つのメモリ装置14aおよび14bに接続する。メモリ装置14aおよび14bは、サーバのためのローカルメモリ記憶装置であってもよいし、サーバ14に結合されている遠隔記憶装置であってもよい。各ディスク駆動システムは、サーバがアクセスおよび制御できる実行可能なアプリケーションプログラムコードのディレクトリおよびファイルを含むコンピュータファイルのセットを保持できる。サーバ14は、スタンドアロンコンピュータシステムであってもよいし、あるいは図示されるように、サーバは、ローカルメモリ装置あるいはネットワークメモリ装置にアクセスできる1つあるいはそれ以上のデータプロセッサを有するネットワークコンピュータシステムであってもよい。

図1は、本発明のシステムがWebサイト管理者が遠隔アプリケーションに結合するWebページを提供できることによってWWWの機能を拡張することを示している。さらに、Webサイト管理者は、クライアントサイトのアプリケーションプログラムのトランスペアレント導入をもたらすために本発明のシステムを使用できる。これによって、Webサイト管理者は、ブラウザあるいはランチャプログラムによって会社のイントラネット内部および会社領域にわたって実行するのに利用されるレガシーアプリケーション、スタンドアロンアプリケーションおよびクライアント/サーバアプリケーションを含むアプリケーションを作成できる。これは、通信要素18、メモリ装置14aおよび14bおよびLAN13bに接続するサーバ14によって図1に示されている。例えば、ユーザがミューチュアルファンドに投資できるアプリケーションプログラムのようなある種のアプリケーションプログラムへのアクセスをユーザに提供することを課されたネットワーク管理者は、メモリ装置の1つ14aあるいは14bにアプリケーションプログラムを記憶できる。サイト管理者は、ノード19aあるいは19bのユーザがアプリケーションプログラムにアクセスし、ファンドを投資および管理するためのプログラムを実行できるサーバ処理をサーバ14にインストールできる。したがって、サイト管理者は、このアプリケーションプログラムをLAN13bに接続されたユーザに供給できる。本発明のシステムおよび方法によって、サイ

ト管理者は、遠隔クライアントにアプリケーションプログラムへのアクセスを提供することができる。例えば、HTTPサーバ処理はサーバ14にインストールできる。HTTPサーバ処理は、HTMLページ内部のリンクに関連し、アプリケーションプログラムを実行するのに関連している要素、環境および他のデータを示す情報を記憶できるアプリケーション情報ファイルに遠隔クライアントを結合するHTMLページ内部のリンクを提供する。クライアントのブラウザ16は、16aのようなグラフィック装置としてリンクを示すHTMLページを表示する。クライアントは、遠隔アプリケーション要素を含む共有ファイルシステムを備え、このアプリケーションを実行するのに適切な環境を作成し、アプリケーションを呼び出すことができる。アプリケーションは、クライアントメモリ内部で局所的に実行できる。アプリケーションの終了と同時に、クライアントは、HTTPサ

ーバ処理に通知し、このサーバ処理はクライアントにもはやサービスをしない。

図1は、クライアント12がローカルエリアネットワーク13aによって接続されるサーバ15のようなローカルサーバによって制御されるプログラムおよびファイルにもアクセスできることをさらに示している。特に、本発明のシステムは、HTTPサーバ14のような遠隔サイトに記憶されたアプリケーションだけでなく、LAN13aで実行するイントラネットのような会社ネットワークによって提供されるアプリケーションプログラムをクライアントが実行できるように既存のLAN技術に組み込むことができることを図1は示している。したがって、本発明のシステムおよび方法を既存のイントラネット技術と結合することによって、イントラネットのノードは、遠隔サイトに記憶されたアプリケーションプログラムにアクセスでき、その結果、本発明のシステムは、これらの遠隔サイトのアプリケーションプログラムへのトランスペアレントアクセスを提供する。

図2は、Webページアプリケーションリンクを生じる1つの処理を示している。この処理では、アプリケーションプログラムは共有ディレクトリあるいはエクスポートファイルシステム内部に備えられる。このディレクトリは、LANマネージャシステム、NFSシステム、NCPシステムあるいは任意の他の適当な

ファイルシステムサーバのようなファイルシステムサーバによって共有されるべきであるか、あるいは任意にはHTTPリダイレクタを使用することによってHTTPサーバに配置できる。共有ディレクトリのアプリケーションプログラムに関しては、ステップ21によって示されるようにWebサイト管理者は、遠隔アプリケーションのためのアプリケーション情報ファイルを生成できる。任意には、ファイルは、サーバで実行する処理によって自動的に作成することができる。

このアプリケーション情報ファイルは、クライアントで遠隔アプリケーションを実行するのに必要な情報を含んでいる。例えば、アプリケーション情報ファイルは、実行可能なファイル（すなわち、EXEファイル）、初期化ファイル（すなわち、INIファイル）、システムファイル（すなわち、SYSファイル）、ディレクトリ構造情報、およびアプリケーションを実行するのに適切な環境を識別する任意の他の情報あるいは構成パラメータを含んでもよい。クライアントで作動する処理は、アプリケーションプログラムを適切に実行できるようにオペレ

ーティング環境を適切に構成する。

アプリケーション情報ファイルが作成されると、ステップ23の処理は、アプリケーション情報ファイルのリンクを示すアイコンを、サーバによって送り出されたHTMLページ内部に埋め込む。任意には、アプリケーションリンクのアクセス規則セットは規定されてもよい。これは、一般的にはアプリケーションリンクの展開中に行われ、リンク先のアプリケーションを実行出来るユーザあるいはクライアントのグループを定義し、また、ユーザに課金する料金に応じて、同時に実行できるアプリケーションの数を定義したアクセス規則など、各種のアクセス規則が設定されてもよい。アクセス制御が整えば、アプリケーションプログラムは呼び出されて実行される。

前述の記載から、図1および図2に示されたシステムおよび方法は、市販のハードウェアで作動でき、一般的には、ネットワーク13bのような既存のLAN上にサイト管理者によってインストールすることができることが明らかである。例えば、各メモリ装置14aおよび14bは、市販のハードディスクシステム、あるいはハードディスクシステムの一部、RAMカードシステム、あるいはコン

ビュータRAMメモリであってもよい。しかしながら、データを記憶し、データプログラムにインタフェースするのに適している任意のメモリ装置あるいはコンピュータネットワークはその範囲から逸脱しないで本発明で実行できることがコンピュータ工学技術の当業者に明らかである。

通信要素18は、クライアント12とサーバ14との間にテレコミュニケーションリンクを保持するモデムとして象徴的に示されている。しかしながら、通信要素も、ISDN装置、光ファイバ装置、ケーブル装置あるいは遠隔サイトに通信リンクを確立するのに適している任意の他の種類の通信装置であってもよい。ローカルエリアネットワークは、イーサネットシステム、あるいはデータ処理システム間でデータを伝達するのに適している任意のネットワークシステムであってもよい。

図3は、図1のシステム10をより詳細に示し、クライアント12がサーバ14に記憶されたアプリケーションプログラムを実行するように要求する前の状態のシステム10を示している。図3は、クライアント要素12と、サーバ要素1

4と、操作コレクション要素20と、ファイルシステム22と、デバイスドライバ26と、ディスク取り付け要素28と、ディスク取り外し要素30と、構成・管理要素32と、データトランスポートインタフェース34と、伝送パス36と、第2のデータトランスポートインタフェース要素40と、ファイル共有システム42と、管理プログラム44と、区分の中の少なくとも1つがファイルシステム56のような論理ファイルシステムを含んでいる、PC区分50、Unix区分52およびMac区分54を有するメモリ要素46とを含み、サーバ14にアクセスできるファイル階層構造58を含んでいるシステム10を示している。

図3は、システム10の要素としての各機能ブロックを示している。しかしながら、各要素は、システム10あるいはシステム10のサブシステムとして示されている装置として作動するようにデータ処理装置を構成するデータ処理装置で作動するソフトウェアルーチンであってもよいことはコンピュータおよびソフトウェア工学技術の当業者に明らかである。さらに、図示された機能ブロックがシステム10の機能を実行するように作動するハードウェアおよびソフトウェアの

組み合わせられた要素であってもよいことはコンピュータおよびソフトウェア工学技術の当業者に明らかである。

図3に示されるように、クライアント12は、一般的にはインターネットサイトあるいはイントラネットサイトに接続を確立するのに適しているTCP/IPインタフェースであるデータトランスポートインタフェース要素34を含んでいる。ネットスケープインターネットブラウザのようなクライアント12で実行するブラウザプログラム処理は、TCP/IPデータトランスポートインタフェース要素34とインタフェースし、TCP/IPプロトコルの下でサーバ14にデータを転送する。同様に、システム10の操作コレクション要素20、取り付け要素28、取り外し要素30、および他の要素は、TCP/IPデータトランスポートインタフェース要素34を使用し、クライアント12とサーバ14との間でデータおよびコマンドを転送する。インタフェース要素34は、Trumpet Winsock TCP/IPデータトランスポートインタフェースプログラムのようなWindowsオペレーティングシステムプログラムをインターネットにインタフェースするWinsockプロトコルに従う任意のプログラムのよ

うな任意の従来のインターネットプロトコルインタフェースであってもよい。しかしながら、IPネットワークでデータを移送させるのに適しているいかなるシステムもその範囲を逸脱しないで本発明で実行できることはコンピュータ工学技術の当業者に明らかである。

図示された実施例において、サーバ要素14も、伝送パス36を介してクライアント要素12のデータトランスポートインタフェース34に接続するTCP/IPデータトランスポートインタフェース40を含んでいる。このデータトランスポートインタフェース40は、サーバ14間でデータおよび制御情報を移送し、それによってファイル共有システム42および管理プログラム44を支援するトランスポート層を提供する。統合トランスポート層プロトコルを含むサーバの開発はコンピュータ工学技術で周知であり、クライアント12とサーバ14との間でのデータの交換を可能にする任意のトランスポートプロトコル層は、本発明

で実行するのに適しており、その範囲内にあるとみなされる。

図3によってさらに示されるように、図示されたサーバ14はサーバ処理48を含んでいる。サーバ処理48は、クライアント12をサーバ14に接続するようにデータトランスポートインタフェース40にインタフェースする。本発明の1つの好ましい実施例において、サーバ処理48は、システムの他の要素だけではなく、クライアント12からのハイパーテキストリンク要求に応じてHTMLページをクライアント12に供給するWebサーバ処理である。サーバ処理48は、本発明の処理を実行し、コンピュータおよびソフトウェア工学技術の公知の原理により構成されるプログラムである。

図3にさらに示されるように、サーバ14は、実行可能なアプリケーションプログラムをクライアント12に供給するサーバ14を構成する管理プログラム44を含んでいる。1つの実施例において、管理プログラム44は、メモリ装置46をその各々がデバイス名を有してもよく、その各々が論理ファイルシステム56を含んでもよいセクション50、52および54として図示されるいくつかの論理セクションに分割する分割プログラムを含んでいる。このような分割プログラムの構造はソフトウェア工学技術において周知である。

図3の実施例において、分割プログラムは、Unixファイルシステムプロト

コルによりデータを記憶する論理セクション52を形成する。図示されるように、区分52は、ファイルシステム56がブートブロック、スーパーブロック、ディスク46上のデータの物理ロケーションを示すデータファイルブロックのセットのポインタの配列を示すinodeリストを含む1つのファイルからなる論理ファイルシステム56を含んでいる。ファイルシステム56は、従来のUNIXオペレーティングシステムファイルシステムであってもよい。図2によってさらに示されるように、各論理ディスク分割セクションは、DOS、UNIX、VAX、あるいは各論理ディスクセクションが、関連ディスク分割のファイルに記憶されている物理ディスクロケーションを示すローレベル情報をプログラムローダに通信するデバイスドライバに供給する論理ファイルシステム56を含む任意の他のファイルシステムプロトコルのような異なる論理システムファイルプロト

コールにより異なる種類のコンピュータファイルに記憶する管理プログラム14によって使用することができる。

本発明の1つの実施例において、管理プログラム44は、インターネットサイトに接続できる任意の種類のクライアント12で実行するアプリケーションプログラムを含んでもよいコンピュータファイルのファイルシステムを提供するように各論理セクション50、52、および54を作成する分割サブルーチンを含んでいる。このようなサブルーチンは当該技術分野で周知であり、任意の適当なサブルーチンあるいはオペレーティングシステム呼び出しは本発明によって使用することができる。特に、管理プログラムは、ディスク46を各異なる種類のクライアント12に対する論理セクションに細分割するために使用できる。一般的には、クライアントのタイプは、クライアント12によって使用されるプロセッサのタイプだけではなく、クライアントが実行するオペレーティングシステムのタイプによって決定される。各クライアント12は、サーバ14がディスク46のどの論理セクションがこの特定のクライアントに適しているファイルを保持しているかを決定できるクライアントタイプ信号をサーバ14に供給する。例えば、Unixオペレーティングシステムに基づいて作動するクライアント12は論理セクション52の下に記憶されたファイルにアクセスでき、DOSを実行する486PCコンパチコンピュータは論理セクション50にアクセスする。本発明の

方法を実行する管理プログラム44は、UNIXオペレーティングシステムのようなサーバオペレーティングシステムにシステムコールを使用するコンパイルされたC++ソフトウェアモジュールのようなソフトウェアモジュールとしてソフトウェア工学の周知の原理により構成することができる。しかしながら、管理プログラムの機能を実行するいかなる構造もその範囲を逸脱しないで本発明で実行できることはコンピュータおよびソフトウェア工学技術の当業者に明らかである。

本発明の他の実施例において、管理プログラムは、クライアント特権信号の機能としてクライアント12に使用可能なアクセス権を決定するファイルアクセス制御プログラムを含んでいる。1つの実施例において、ファイルアクセス制御プ

ログラムはクライアント識別信号のデータベースを含んでいる。本実施例では、管理プログラム32は、クライアント12を独自に識別するクライアント識別信号を供給するクライアント管理プログラムを含んでいる。ファイルアクセス制御プログラムは、クライアント識別信号のデータベースを含み、各クライアント識別信号に対するクライアント特権フィールドをさらに含んでいる。ファイル制御プログラムは、クライアント特権フィールドを使用し、クライアント12のファイルアクセス権を決定する。ファイルアクセス制御プログラムは、クライアント12に使用可能にするように適切なファイルシステムあるいはディスク分割を決定できる管理プログラム44にファイルアクセス権を提供する。

したがって、管理プログラムはシステム10のためのライセンス・セキュリティシステムを供給できる。パスワードシステム、オンライン課金システム、あるいは、ファイルアクセス権を制御するシステムなど、サーバファイルシステム階層58に取り付けられたファイルを制御する、あるいは、クライアント12に提供されたサーバファイルポインタのコピーを制御するファイルアクセスコントロールプログラムは、その範囲から逸脱しないで本発明で実施できることは当業者に明らかである。管理プログラム44は、管理者、あるいは任意にはクライアントがWebページのリンクとして指定することができるアプリケーション情報ファイルを作成および構成できる処理も含んでもよい。他の実施例において、管理プログラム44は、システム使用を監視し、サイト管理者がリソースを追跡し、制御できる統計を生成する処理を含んでもよい。

本発明の他の実施例において、管理要素32は、クライアントタイプ信号とともに、クライアントの特徴を示す情報信号を有するクライアントファイルをサーバ14に提供するクライアント管理要素を含んでいる。これらの特徴は、クライアントのメモリサイズと、使用可能なプログラムメモリと、config.sysファイル、win.iniファイル、ローカルアプリケーションのための構成ファイルを含むシステム構成と、クライアント12に記憶されたシステムの1つあるいはそれ以上のファイル名とを含んでもよい。管理プログラム44は、他の従来の管理機能に関してだけでなく、どのファイルをクライアントに提供するかを決定するために

この情報を使用する。

図示された実施例において、管理プログラム44はサーバ処理48から分離し、しかしながら、管理プログラムがサーバ処理48の一部として実行できることはコンピュータおよびソフトウェア工学技術の当業者に明らかである。

図3に示されるファイル共有システム42は、メモリ装置46と通信し、サーバ要素14がメモリ装置46間でコンピュータファイルのような情報を読み出しおよび書き込みできるようにする。ファイル共有システム42は、サーバ14のためのファイル階層58をさらに保持し、それによってメモリ装置46内部のデータブロックとして記憶されるコンピュータファイルをアクセスし、管理する論理構造を提供する。

ファイル共有システムは、ディスクの特定セクションのファイルシステムを既存のサーバファイルシステム階層58の中に接続する取り付け手順42を含んでいる。さらに、取り付け手順呼び出し42は、オペレーティングシステム特定パス名をファイル操作に変換できる。当該技術分野で周知のようにファイル操作は、サーバ14がクライアント12と通信するために使用できるデータアドレス構造である。ファイル操作は、ファイルシステムのファイルのファイルポインタの役目を果たす。一般的には、ファイル操作、あるいはファイルポインタは、ファイルが最初に開かれる場合、サーバ14のためにファイルを識別し、サーバ14が操作によって示されるファイルをアクセスするために使用できる操作をその後提供するシステム独特の32ビット数である。1つの実施例において、ファイル共有システム42は、Unixプラットフォームでファイルシステムを共有するフ

イルポインタのノードアレイを形成する、カリフォルニア州のサンマイクロシステムズによって開発されたNFSシステムである。しかしながら、ファイルシステムを取り付け、ファイルシステムのファイルのファイルポインタの役目を果たすファイル操作を生成できる任意のファイル共有システムはその範囲を逸脱しないで本発明で実施できることは当業者に明らかである。

クライアント12はデバイスドライバ26を含んでいる。図示されたデバイス

ドライバ26は、ダイナミック取り付け手順28、ダイナミック取り外し手順30および1つあるいはそれ以上の構成・管理手順32を制御する。下記により詳細に記載されているように、インターネットサイトに記憶されたアプリケーションプログラムをアクセスするために、デバイスドライバ26は、ダイナミック取り付け手順28を起動する。取り付け手順28は、データトランスポートインタフェースに送られ、サーバ14に移送される遠隔手順呼び出しを形成する。

クライアント12が実行するためのアプリケーションプログラムを選択した後の状態でシステム10を示す図4に関して、システム10はさらに説明できる。図4に示されるように、システム10は、図3の要素およびファイルポインタ60のサーバアレイの他の要素、ファイル階層58のクライアント取り付けポイントに取り付けられた論理ファイルシステム56とクライアント12のファイルシステム22によって保持される遠隔ファイルポインタ24のアレイを含んでいるファイル階層58の他の要素を含んでいる。システム10は、論理ファイルシステム56内部に記憶されたアプリケーションプログラムを実行することを要求するクライアント12に応じてファイルポインタ60のサーバアレイおよび遠隔ファイルポインタ24のアレイを生成する。

一実施例において、クライアント12は、Webブラウザプログラムを使用し、サーバ14で実行するサーバ処理48に接触する。ブラウザプログラムは、サーバ処理48のためにクライアント12のシステムタイプおよびクライアントシステム12によって使用されるファイルシステムプロトコルを示すクライアントタイプ信号をサーバ処理48に供給する。

図示された実施例において、サーバ処理48は、管理プログラム44にクライアントタイプ信号を供給する。管理プログラム44は、クライアントタイプ信号

を使用し、クライアント12に供給されるべきである互換性論理ファイルシステムを決定する。一実施例において、管理プログラム44は、クライアント12に供給される適切なファイルシステムを決定し、クライアント12と互換性があり、クライアント12にアクセス可能にされるファイルシステム56のディレクトリ構造を示すディレクトリ情報信号を含むURLドキュメントのポインタをサー

サーバ処理 48 に供給する。管理プログラム 44 は、ディレクトリ情報信号を図形により描写するハイパーテキストページを提供する HTTP URL を Web サーバ 48 に供給できる。任意には、管理プログラム 44 は、論理ファイルシステム 56 のディレクトリ構造のより詳細を提供する他の URL ドキュメントのハイパーテキストリンクであるグラフィックアイコンとしてディレクトリ情報信号を示すハイパーテキスト信号を生成するハイパーテキストリンクプログラムを有するページ要素を含んでいる。

サーバ処理 48 は、使用可能な論理ファイルシステム 56 のディレクトリ情報をクライアント 12 のために移送し、サーバ 14 のインターネットネットワークアドレスおよびファイルシステム 56 のパス名を含むデータファイルをさらに供給する。

接続要素 20 は、サーバ処理 48 によって供給されたデータファイルを受け取り、ネットワークアドレス信号およびパス名信号を取り出す。コレクション要素 20 は、ネットワークアドレス信号およびパス名信号を使用し、遠隔ファイルポインタ 24 のアレイを生成し、システム 56 のようなファイルシステムを取り付けるのに必要なアーギュメントを取り付け手順 28 に提供する。一般的には、接続要素は、連結プログラムを含み、適切な取り付けポイントを生成するようにアドレスおよびパス名情報を連結する。他の実施例において、接続要素は、管理プログラムから適切な取り付けポイントおよびファイルシステム名を受け取る。この処理の他のステップにおいて、クライアント 12 のブラウザプログラムは、論理ファイルシステム 56 のグラフィック描写をクライアントユーザに対して表示する。

本発明の一実施例において、クライアントユーザは、アプリケーションプログラムでクリックすることによってファイルシステム 56 内部で識別されたアプリ

ケーションプログラムを実行するように要求することができる。ユーザの動作に応じて、接続要素 20 は、必要なアーギュメントを送り、クライアント 12 によって要求されたアプリケーションプログラムを含む論理ファイルシステム 56 をファイル階層 58 に取り付けるようにサーバ 12 に命令する遠隔手順呼び出しを

生成する取り付け手順28を起動する。

一実施例において、接続要素20は、サーバのIPアドレスおよびクライアント12に使用可能なファイルのパス名を提供するサーバ処理48からデータファイルを受け取る。他の実施例において、接続要素20は、サーバ14のアドレス情報を供給する、周知のポート80のようなサーバ処理のポートの呼び出しを開始する。一例において、クライアント12は、アプリケーションプログラムXLANDER.EXEを実行することを選択できる。一旦コレクション要素がネットワークアドレスおよびパス名を決定すると、接続要素20は、ファイルXLANDER.EXEを含むファイルシステムを動的に取り付けるために取り付け手順を起動できる。1つのこのような取り付け手順28は下記の擬似コードから理解される。

```
int mount fs(mntp)
struct mntent *mntp;
{
    int error;
```

/*取り付け手順は、MOUNT - TRAP呼び出しを行う。この呼び出しは、オペレーティングシステムライブラリの呼び出しであるMOUNT - TRAPルーチン呼び出す。WinライブラリあるいはSYSライブラリによって、1例としてこのような関数が与えられる。あなたもあなた自身の取り付け関数を記述できる。

```
error=MOUNT_TRAP(type,mntp,flags,mnt_data);
if(error<0){
    plog(XLOG_ERROR, "mount failed on %s:errno=%d",
```



```

        mntp->mnt_dir,ermo);

    return(-1);
}

}

main(argc,argv)

int argc;
char**argv;
{

/*メインルーチンは、コレクションプログラムからホスト名および取り付けポイ
ント名を受け取る*/

    hostname=argv[1];

    mp_name=argv[2];

/*メインルーチンは、それからWebサーバ処理のためのファイル名をフォーマ
ット化するように備えられるべきファイルシステムの名前を再フォーマット化す
る。一般的には、ファイル拡張“.wfs”は、WWWファイルシステムを示す
ために付加される*/

    fs_name=compute_fs_name(argv[1]);

/*メインルーチンは、それからファイルシステム名、取り付けポイント名、取り
付けタイプフィールドを1つの構造にパックする。*/

    mntp->mnt_dir=mp_name;

    mntp->mnt_fs_name=fs_name;

    mntp->mnt_type=MNTTYPE_WWFS;

/*発するメインは、mntp構造で作動する取り付けファイルシステム手順を呼び出
す*/

    ret=mount_fs(&mntp,flags,(caddr_t)&nfs_args,1,MOUNT_TYPE_NF
S);

    return ret;
}

```

上記の擬似コードによって示されるように、取り付け手順28は、コレクション要素30から、取り付けるためのファイルシステムおよびファイルシステムのための取り付けポイントを示す2つのアーギュメントを受け取る。取り付け手順28は、それから、取り付け手順62を起動するためにファイル共有システム42のための遠隔手順呼び出しを形成できる。取り付け手順62は、XLANDER.EXEアプリケーションプログラムのための実行可能なプログラムファイルを含んでいるファイルシステム56のようなファイルシステムをサーバファイル階層58の中に取り付ける。次に、取り付けられたファイルシステムはアクセスできる。

ファイル共有システム42は、XLANDER.EXEのための実行可能なコードを含んでいるディスク46のロケーションを示すファイルptr 1 - nとして図示されるファイルポインタ60のアレイを形成するプログラム要素である。しかしながら、ファイルポインタ60のコピーをクライアントに供給するいかなるシステムもその範囲を逸脱しないで本発明で利用できる。

図5に示されるように、接続要素20は、ファイルポインタ60のコピーを有するファイルを開き、遠隔ファイルポインタ24のアレイを生成する。IP+PATH+FILEPTRとして図示される各遠隔ファイルポインタ24は、取り付けられたディスク分割による操作、およびXLANDER.EXEのための実行可能なコードを含むディスク46の物理ロケーションのアクセスポイントとしてクライアント12によって使用できる取り付けられたディスク46の物理ロケーションに対する操作である。本発明の一実施例において、遠隔ファイルポインタのアレイは、ファイルポインタのアレイをクライアント12のファイル階層に統合することによってクライアント12に供給される。遠隔ファイルポインタをファイルシステムに供給するのに適しているいかなる接続要素も、ストリングWWW.EPICON.COM:USR/LOCAL/STORE/XLANDER.EXE（ここで、WWW.EPICON.COMはIPアドレスを示すサーバアドレスであり、USRはディスク46を示し、および/LOCAL/STORED/XLANDER.EXEはパス名信号を示している。）を形成する接続要素のようなサー

バアドレス信号をパス名信号およびファイルポインタ信号を連結することによってポインタを形成する連結ルーチンを使用する接続要素を含む本発明で実行できることが理解される。

図5は、クライアント12がプログラムローダ要素62を含んでもよいことをさらに示している。図5に示されたプログラムローダ要素は、デバイスドライバ26に接続し、クライアント12のユーザによって選択された実行可能なファイルにアクセスするようにデバイスドライバ26を使用する。デバイスドライバ26は、クライアント12のファイル階層を探索し、選択された実行可能なプログラムのコードの最初のブロックを指す遠隔ファイルポインタを識別する。データトランスポートインタフェース34にインタフェースするデバイスドライバ24は、ディスク46からのファイルブロックの転送を要求することによって実行可能なコードのファイルブロックにアクセスする。プログラムローダはブラウザ処理におけるウィンドウ内部の処理を実行できる。例えば、一実施例において、クライアント12は、WINDOWS 95オペレーティングシステムを実行するように構成され、このオペレーティングシステムの下で供給されたプログラムローダを使用した。しかしながら、実行可能なコードをロードするファイルポインタを使用するのに適しているいかなるプログラムローダもその範囲を逸脱しない本発明で使用できることは明らかである。

図5は、パス36にわたって転送されるデータを圧縮する圧縮プログラム38をさらに示している。本実施例において、サーバ処理48もデータトランスポート層によって伝送されるデータを圧縮し、伸長する圧縮ルーチンを含んでいる。一実施例において、本発明は、ハフマン符号化手順を実行する圧縮プログラムを提供する、しかしながら、他の符号化処理の取り換えは本発明の範囲から逸脱しないことは当業者に明らかである。他の好ましい実施例において、システム10は、サーバ14がディスク46から読み出され、クライアント12に移送するために用意されているデータファイルブロックをキャッシュできるキャッシングシステムを含んでいる。キャッシングシステムは、クライアント12に転送するためのデータブロックのバッファを高アクセス速度メモリに記憶する任意のキャッシングシステムであってもよい。

図6は、キャッシュメモリシステムと、HTTPリダイレクタと、キャッシュリダイレクタとを含んでいる本発明による他のシステムの構成要素および相互作用を図形により示している。特に、図6は、クライアント要素212と、HTTPサーバ214と、キャッシュシステム216と、ファイルシステムサーバ218と、ブラウザ222と、ブラウザインタフェース224と、ランチャプログラム226と、ファイルシステム接続処理228と、遠隔アクセス接続230と、ネットワークAPI234と、リダイレクタAPI236と、ファイルシステムキャッシュAPI238と、オペレーティングシステムファイルシステムインタフェース240と、キャッシュリダイレクタ242と、ネットワークファイルシステムクライアント250と、オプションNFSリダイレクタ252と、HTTPリダイレクタ254と、他のリダイレクタ258と、ミニリダイレクタカーネル要素260とを含んでいるシステムを示している。

ブラウザ222は、前述のようなブラウザであってもよく、クライアントインタフェース224は、ネストスケーププラグイン、アクティブX処理、あるいはブラウザプログラム222をクライアント212の他の要素とインタフェースする任意の同様な処理のようなインタフェース処理であってもよい。

同様に、ランチャプログラム226は、アクティブXアプリケーション、プラグインあるいはブラウザプログラム222によって起動できる任意の他のコンピュータ処理であってもよい。示されたランチャプログラムは、サーバ214から起動アプリケーションに関連したアプリケーション情報ファイルを検索し、遠隔アプリケーション構成要素を含む共有ファイルシステムを取り付け、アプリケーション情報ファイルの情報に応じて、アプリケーションを開始および実行するためにクライアント212内部に選択環境を形成できるコンピュータ処理である。ランチャプログラム226は、アプリケーション実行を監視し、アプリケーションが終了するとサーバ214に知らせる。

ランチャプログラム226は、遠隔アプリケーションあるいはキャッシュアプリケーションのいずれかを実行してもいい。この目的のために、図示されたランチャ226は、ランチャがクライアントファイルシステムにインタフェースし、遠隔ファイルシステムをクライアントのファイルシステムに取り付けることがで

きるファイルシステム接続処理228にインタフェースする。ファイルシステムを備えるこのような処理が前述されている。同様に、ランチャプログラム226は、ランチャがクライアント212によって実行される遠隔アプリケーションにアクセスできる遠隔アクセス処理230とインタフェースしてもいい。遠隔アクセス処理230は実行可能なコードのブロックを動的に送出できる。このコードは、前述のように、クライアントプログラムメモリにロードし、任意には、サーバ214のディレクトリ構造に反映するディレクトリ構造のキャッシュメモリ216内部に記憶できる。この理由で、遠隔アクセス処理230は、キャッシュメモリ216内部のデータの記憶および検索を制御するためにキャッシュAPI238とインタフェースできる。

図7は、ネットワークからサーバ214によって示されたWebページを通る遠隔アプリケーションを実行するのに加えて、アプリケーションが、オフラインページのリンクを任意に起動することによってキャッシュメモリ216からも実行できることを示している。オフラインページは、クライアント212に局所的に記憶されたHTMLページであってもよく、これらのキャッシュプログラムを開始するためにキャッシュメモリ216に記憶されたアプリケーションのリンクを有してもよい。さらに、クライアントはキャッシュメモリからの自動的に開始するアプリケーションのための処理も含んでもよい。一般的には、この処理はキャッシュリダイレクタ242に含められる。この処理は、同じアプリケーションがサーバおよびキャッシュの両方から利用可能であるかどうかを決定できる。キャッシュをアクセスすることがより有効である場合、この処理はアプリケーションのキャッシュバージョンを開始する。

前述のように、オペレーティングシステムは、遠隔アプリケーションをロードする要求を処理し、遠隔アプリケーションファイルにアクセスするファイルシステム要求を適切なネットワークファイルシステムリダイレクタに送る。これは、要求をキャッシュリダイレクタ242に送ることも含んでもよい。キャッシュリダイレクタ242は、遠隔アプリケーションに関連した形成された仮想ドライブに記憶された遠隔ファイルのある種のファイルシステム要求を遮断する。キャッシュリダイレクタは、要求ファイルがキャッシュされたかどうかを決定できる。

要求されたファイルがキャッシュされた場合、キャッシュリダイレクタ242は、ローカルキャッシュメモリ216のファイルのファイル属性と遠隔ファイルの属性とを比較する。属性が遠隔ファイルが変更されたことを示すならば、キャッシュリダイレクタはキャッシュコピーを削除する。

要求されたファイルがキャッシュメモリ216から取り込むことができない場合、キャッシュリダイレクタ242は、この要求を下にあるネットワークファイルシステムクライアントの1つに送る。それから、キャッシュリダイレクタ242は、このファイルを記憶するためキャッシュメモリのスペースを探し、OSファイルシステムが共有ネットワークファイルシステムからファイルを読み出すときにキャッシュされるように当該スペースにマークを付ける。ファイルを収集後、キャッシュリダイレクタ242は、ファイルシステム要求に応じてファイルをOSファイルシステムに戻す。

アプリケーションが終了後、ランチャ処理226は、クリーンアップを実行し、ドライブ/ユニバーサル・ネーミング・コンベンション (UNC) 接続を切り離し、環境を最初の状態に戻し、完全にキャッシュされた構成要素を有する全ての遠隔アプリケーションについてキャッシュリダイレクタ242に照会する。オフラインページを更新し、オフラインアプリケーション情報ファイルをインストールディレクトリにコピーし、任意の新規エントリのためのリンクをオフラインページに追加および/またはファイルがキャッシュ216から追い出した任意のファイルを除去する。クライアント212は、サーバ214にアプリケーション終了事象、あるいは実行失敗事象を知らせる。それから、サーバ214は、アプリケーションに要求される任意の必要な計測管理を実行できる。

前述の情報に加えて、アプリケーション情報ファイルはオフラインページのための情報を含んでもよい。これはオフライン許可フラグを含んでもよい。このフラグがセットされないならば、ランチャ226は、このアプリケーション情報ファイルによって示された遠隔アプリケーションをオフラインページに追加しようと試みない。Web管理者がアプリケーション情報ファイルを作成あるいは更新する場合、このフラグは、サーバ管理者プログラムによってセットすることができる。この情報は、アプリケーションオフラインを実行するようにキャッシュ2

16にキャッシュされるべきである遠隔アプリケーションの構成要素のリストも含んでもよい。Web管理者がアプリケーション情報ファイルを作成する場合、このリストはサーバ管理プログラムによってもセットできる。任意には、リストはオフラインプリロード要求フラグを含んでもよい。このフラグは、ユーザがオフラインページのアプリケーション情報ファイルによって示された遠隔アプリケーションを追加するように要求したことをランチャ226に指示するようにセットされる。ランチャ226は、遠隔アプリケーションの構成要素の全てをキャッシュメモリ216にプリロードし、オフラインページを更新するが、遠隔アプリケーションを開始しない処理を含んでもよい。プリロード処理は、選択アプリケーションプログラムの適切なファイル要求を生成し、ファイル要求をHTTPリダイレクトに送ることによって作動できる。

一実施例において、ランチャは、本発明によりサーバ処理を実行する全ての“既知”Webサーバの階層を示すことができる。ランチャは、これらのサーバの各々の“既知”アプリケーションリンクのリストを作成できる。ランチャが実行を開始すると、ランチャは“既知”サーバおよびそのアプリケーションリンクのリストを得る。サーバのリストは、本発明により他のサーバ処理を知っているようにセットアップされるプロキシサーバに照会要求を送ることによって得ることができる。したがって、アプリケーションリンクのリストは、照会要求を特定サーバに送ることによって得ることができる。照会要求から戻される全てのアプリケーションリンクは、その時、ランチャのウィンドウに表示できる。

ランチャ処理は任意の適当な方法で符号化することができるけれども、実施例は、Windows95およびWindowsNT3.51/4.0のプラットフォームのための32ビットアプリケーション、Windows3.1のための16ビットアプリケーション、UNIX、OS/2、およびMACプラットフォームで作動するようにポータリングされたプロセスであってもよい。

図6に戻ると、キャッシュメモリシステム216は、アプリケーションがアプリケーションリンクをクリックすることによって実行される場合、遠隔アプリケーションおよび関連ファイルをキャッシュするかあるいは反映するクライアントのローカルディスク上のデータベースの役目を果たすことができる。このデータ

ベースは、クライアント導入処理ディレクトリの真下の隠しディレクトリに置かれてもよい。ファイルはサーバのディレクトリ構造とは異なるフォーマットでキャッシュに記憶できるので、ローカルユーザによって直接に解説することができない。キャッシュ216は、遅いリンクを介して遠隔アプリケーションを実行する必要性を避けることによって性能を改善できる。同じリンクを実行する次の要求と同時に、アプリケーションおよび関連ファイルはネットワークを介してよりもむしろキャッシュから読み出すことができる。さらに、キャッシュによって、完全にキャッシュされたアプリケーションは、アプリケーションを含む遠隔ファイルサーバにネットワーク接続が全然ない場合、オフラインページから実行することができる。

図示されたキャッシュリダイレクタ242は、オペレーティングシステムの中央ファイルシステム制御装置と導入されたネットワークファイルシステムとの間にある。このリダイレクタは、ファイルキャッシュ機構を実現するために異なるファイルシステムサービス、NFS、HTTP等を接続する。この層に送られた全てのファイルシステム要求は適当なネットワークファイルシステムに送られる。

この目的のために、ネットワークファイルシステムクライアント250は、クライアントから遠隔I/O要求を受け取り、この遠隔I/O要求を処理するために適切なサーバに送る多重プロトコルファイルシステムとしての役目を果たすことができる。図示された実施例において、ネットワークファイルシステムクライアント250は、ファイル、プリンタ、直列ポート、名前付きパイプ、メールスロット、あるいは他の装置もしくはアブストラクトの遠隔I/O要求を受け取り、この要求をサーバに送る。図示されたように、ネットワークファイルシステム250の各リダイレクタは、ファイル、データあるいは装置を共有する選択されたタイプのプロトコルと併用するためにI/O要求を再指令できる。

ネットワークファイルシステムクライアント250は、遠隔I/O要求をサービスする複数のリダイレクタの中の1つを選択するコントローラ処理を含んでもよい。このコントローラは、I/O要求から要求データあるいは装置にアクセスするサーバを決定することによってリダイレクタを選択できる。さらに、コント

ローラは、複数のサーバから利用されるアプリケーションプログラム、あるいは

他のデータあるいは装置を識別するリストを含んでもよい。コントローラは、使用可能なファイルシステムあるいはトランスポートプロトコルのような特徴あるいは基準に基づいてアクセスするようにサーバの中の1つを選択できる。例えば、アプリケーションプログラムが両方のタイプのサーバから使用可能である場合、NFSリダイレクタはHTTPリダイレクタを介して常に変更できる。それとは別に、コントローラ処理は、より速いネットワークのリダイレクタを選択するためにクライアント要求に対するネットワークの現応答時間を規定する伝送待ち時間期間を決定できる。コントローラ処理は、複数のリダイレクタをカプセル化し、クライアントオペレーティングシステムのインタフェースの役目を果たすラッパーの一部であってもよい。

NFSリダイレクタ252は、送り出された遠隔ファイルシステムに接続し、NFSサーバによって共有される遠隔アプリケーションを実行する要求を処理するファイルシステムクライアントであってもよい。NFSリダイレクタ252は、Windows 95のVxDおよびWindows NTのFSDとして実現できる。Windows 3.1および作業グループのためのWindowsが支援されるならば、NFSリダイレクタ252はリダイレクタVxDとして実現できる。NFSプロトコル、プロトコル特有のキャッシングおよび全OSプラットフォームに共通である（プロトコルあるいはクライアントリダイレクタ特有機能のいずれかに関係している）他の機能を提供するリダイレクタの中心的な部分は、プラットフォームにわたって移動できる別個のOSに無関係なシステム構成要素として実現できる。この中心的な部分はミニリダイレクタ260として図6に示されている。図示されたNFSリダイレクタ252のシェルは、クライアントオペレーティングシステムを支援するように構成され、クライアントの特定のOSに特有なネットワークファイルシステムラッパーとして実現できる。

図示されたHTTPリダイレクタ254は、アプリケーション情報ファイル内部に供給されている全URLパス名を取り出すネットワークファイルシステムクライアントリダイレクタの役目を果たすことができる。これは、HTTP共有フ

ファイルシステムを提供する。したがって、HTTPリダイレクタ254は、HTTPサーバディレクトリにユニバーサル・ネーミング・コンベンション (UNC)

／駆動接続を保持することによってHTTP共有ファイルシステム内部のファイルにアクセスすることができる。ローカルマシンで実行する遠隔アプリケーションの見地から、HTTPリダイレクタ254によるファイルにアクセスする結果は、任意の他の公知のネットワークファイルシステム (すなわち、マイクロソフトネットワーク、あるいはNFS) によるファイルにアクセスする結果と同じである。HTTPリダイレクタ254は、C++コンピュータ言語で実行されるコンピュータ言語であってもよい。このプログラムは、クライアントオペレーティングシステムによって行われるファイル要求を監視あるいはこのファイル要求に応答できる。このプログラムは、HTTPサーバに記憶されたファイルに対するこれらのファイル要求を処理する。一実施例において、プログラムはファイルシステム要求をHTTPサーバによって処理できるHTTPコンプライアントコマンド信号に変換する。HTTPコンプライアントコマンド信号は、簡単な要求あるいは全要求を含んでもよく、GETコマンド、あるいは任意の他のHTTPコマンドあるいはHTTPサーバと通信するのに適している方法を含んでもよい。このサーバは、URLパス名から生成されたファイル操作によって識別されるファイルをHTTPプロトコルにより転送することによってコマンドに応答する。これは、ネットワーク装置の役目を果たすようにHTTPサーバを指向するHTTP共有ファイルシステムを提供する。HTTPリダイレクタ254の開発はソフトウェア工学技術の周知の原理から追求し、このようなリダイレクタの実際の符号化はコンピュータプログラミング技術の通常の技術の中の1つの技術の範囲内にある。

HTTPリダイレクタ254はキャッシュメモリ216との密接な統合に機能する。他の実施例において、HTTPリダイレクタ254は、全ファイルが直ちに読み出される必要があるよりもむしろファイルの一部を読み出す処理を提供できる。HTTPリダイレクタ254は、遠隔アプリケーションファイルに読み出

しアクセスをする。限られた書き込みアクセスは、例えば、HTTP PUT コマンドを使用することによって提供することができる。

HTTPリダイレクタ254の長所は、リダイレクタ254がインターネットにわたり、かつファイヤウォールを通るアプリケーションリンクを実行する支援

を提供する。周知のファイルシステムクライアントおよびサーバソフトウェアを利用することは、イントラネットにわたるアプリケーションリンクを実行するのに十分であり、しかしながら、インターネットにわたるこれらの周知のファイルシステムサーバの1つにアクセスできることはセキュリティホールを形成する。それとは別に、HTTPサーバは、ネットワークのファイヤウォールに配置できる。インターネットからアクセスできるのか望まれる全アプリケーションはこのWebサーバに駐在すべきであり、アプリケーションリンクはそれに応じてセットアップできる。したがって、アプリケーションはHTTPリダイレクタ254によってインターネットにわたって実行できる。

HTTPリダイレクタ254は、クライアントOSとインタフェースするミニリダイレクタおよびラッパー内部で実行される中心的な部分を有するNFSリダイレクタと同様に実行できる。

本発明のシステムは、プリインストールされるべきアプリケーションにも有益である。例えば、アプリケーションがいくつかのシステムレベル構成要素、例えば、このアプリケーションの導入中OSと統合される必要があるドライバを含むならばあるいは導入がクライアントシステムリブートを必要とするならば、このようなアプリケーションはプリインストールされるべきである。これにアドレスするために、本発明は、クライアントにこのようなアプリケーションの遠隔導入をもたらすので、クライアントは、ローカルマシンに遠隔アプリケーションをインストールするように選択できる。この導入の方法はクライアントマシンの遠隔インストールあるいはセットアッププログラムを実行し、それからインストールプログラムは、クライアントによって取り付けられたネットワークドライブからアプリケーションの必要構成要素をコピーする。

図8は、アプリケーションプログラムのディレクトリ構造を提供するグラフィ

ックインタフェース16を示している。図8に示されるように、システム10は、ブラウザプログラム78のウィンドウ内部に表示するグラフィックユーザインタフェース16を提供する。本発明の図示された実施例において、グラフィックユーザインタフェース16は、Webサーバ処理48によって提供され、サーバ14のファイル階層に取り付けられているファイルシステム56のファイル構造を

示すディレクトリ情報信号を図で示すハイパーテキストリンクを含むHTMLハイパーリンクページである。

図8に示されるように、グラフィックユーザインタフェース16は、ディレクトリ構造、サブディレクトリ構造、ファイル構造、ディスク構造、あるいは他の共通ファイルシステム構造のようなディレクトリプリミティブを記述するグラフィックアイコンとしてハイパーテキストリンクを提供する。図5は、ディレクトリアイコン70と、サブディレクトリアイコン72と、ファイルアイコン74A、74Bおよび74Cとを含んでいる1つのグラフィックファイル構造を示している。本発明の一実施例において、各アイコンは、取り付けられたファイルシステム56のディレクトリ構造に関する他の情報を提供する付加HTMLページをダウンロードするようにWebサーバ処理48を要求する他のHTTP URLのハイパーテキストリンクである。本発明によるシステムおよび方法はこのようなグラフィックインタフェースなしで実行できることは当業者に明らかである。さらに、URLサイトに複数のリンクを提供するHTMLサーバ処理の構成は、コンピュータ工学技術で周知であり、図8に示されたインタフェース16の任意の変更、削除、あるいは追加は、当業者の範囲内にあるとみなされ、それは本発明の範囲を逸脱しない。

図9は、クライアントがそれからのアプリケーションを実行するインターネットサーバに接続できる本発明による1つの処理を示す機能ブロック図を示している。図9に示されるように、本発明による処理は、部分的に互いに無関係に作動する、クライアントコンピュータで実行する第1の処理およびサーバコンピュータで実行する第2の処理の2つの処理として理解できる。図9はこの図の左側の

部分のサーバ処理を示している。サーバ処理はステップ170、180、190および200を含んでいる。クライアント処理は、図9の右側に示され、ステップ110、120、130、140、150および160を含んでいる。

処理100は、クライアントをWebサーバ処理に接続するクライアント処理ステップ110で開始する。図9に示されるように、クライアント処理のステップ110によって、サーバ処理は、ファイルをアクセスするためにWebサーバアクセス要求を受信することを含むステップ170を実行する。サーバ処理は、

サーバ12が、クライアントを認証し、クライアントによってサーバに伝送されるクライアントID信号およびクライアント特権信号を調べることによってそのクライアントに与えられたアクセス特権を決定する任意のステップであるステップ180に続く。任意のステップ180後に、サーバ処理は、ステップ190に続き、クライアントにIPアドレス信号およびパス名信号を戻す。

本発明の好ましい実施例において、ステップ190は、使用可能なアプリケーションプログラムを図で示すHTMLページをクライアントに戻すステップも含んでいる。図4に示されるように、クライアント処理は、エージェントがIPアドレスおよびパス名を受け取るステップ120に進む。IPアドレス信号およびパス名信号の受信と同時に、クライアント処理は、ステップ130に進み、IPアドレス信号を使用し、ダイナミック取り付け要求を発生する。本発明の好ましい実施例において、クライアントは、ダイナミック取り付け要求をサーバ14のWebサーバ処理に供給するハイパーテキストリンクを起動するクライアントによってダイナミック取り付け要求を発生する。

図9によってさらに示されるように、クライアント処理のダイナミック取り付け処理はサーバ処理に送られ、サーバ処理はステップ190からステップ200に進む。ステップ200において、サーバ処理は、クライアントのファイル処理要求を処理し、クライアントによって要求されたアプリケーションファイルを含む遠隔システムディスクの取り付けを実行する。前述のように、処理ステップ200は、アプリケーションプログラムのデータブロックのファイル操作の役目を果たすサーバファイルポインタのアレイを生成するステップを含んでいる。

図9に示されるように、ステップ200は、ステップ140に進み、取り付けられた遠隔ドライブからアプリケーションを呼び出すクライアント処理にサーバファイルポインタのアレイのコピーに送る。ステップ140において、クライアント処理は、IPアドレス、パス名信号、およびサーバファイルポインタの機能としてクライアント処理によって生成されるファイルポインタの遠隔アレイによって指示されたファイルブロックにロードするためにプログラムローダを呼び出すことによってクライアントシステムの通常の実行処理に続く。

図9に示されるように、クライアント処理は、アプリケーションの実行後、ス

テップ140からステップ150に進み、アプリケーションから出る。図9によってさらに示されるように、クライアント処理は、処理100の制御をクライアントの取り外し要求を処理し、取り付けられたドライブをサーバシステムファイル階層から取り除くサーバ処理に転送する任意のダイナミック取り外しドライブ要求を発するステップ160に進むことができる。ダイナミック取り外し要求は、特定のファイルシステム、一般的には、取り付け手順28によって取り付けられたファイルシステムを取り外すためにファイル共有システム42の遠隔手順呼び出しを形成する。取り外し手順30の構成は前述された取り付け手順28と同様である。

図9によってさらに示されるように、処理100は、ステップ150後に作動し続けることができ、クライアント12が実行する次のアプリケーションプログラムを選択できる。ここに使用されている用語としての遠隔アプリケーションは、単一の実行可能なファイル、ならびにアプリケーションを実行する構成要素を含むディレクトリのセットならびにこのアプリケーションを実行するローカルマシンに適用されるべき規則のセット（環境変数、登録および他のシステム構成ファイルの変更）を含むように理解される。

したがって、本発明は、前述の説明から明らかにされた目的の中から上記に詳述された目的を効率的に得ることが分かる。変更が上記の構成および前述のシーケンスおよび動作で本発明の範囲から逸脱しないで行うことができることも分かる。例えば、本発明は、アプリケーションプログラムあるいはプログラミング言

語が遠隔ディスクを取り付け、インターネット接続を介してこのディスクからのアプリケーションプログラムを実行できるアプリケーションプログラムあるいはプログラミング言語に埋め込まれたプリミティブを含むように理解できる。一例として、データトランスポート層に通信する機能性を含むアプリケーションプログラムは、ユーザがアプリケーションプログラム範囲内から、サーバサイトに結合されたディスクに置かれた第2のプログラムをアクセスし、実行できるように本発明のシステムおよび方法を使用できることが分かる。さらに、オペレーティングシステムは本発明のシステムおよび方法を含むように適合できることが分かる。

したがって、添付図面に示された全ての事項はいかなる限定する意味においてよりもむしろ例示として解釈されるべきことが意図される。下記の請求の範囲は、ここに示されているような本発明の一般的な特徴および特定の特徴の全ておよび言葉の問題としてその間にあるとすることができる本発明の範囲の全ての主張をカバーすると意図されていることも分かる。

【图1】

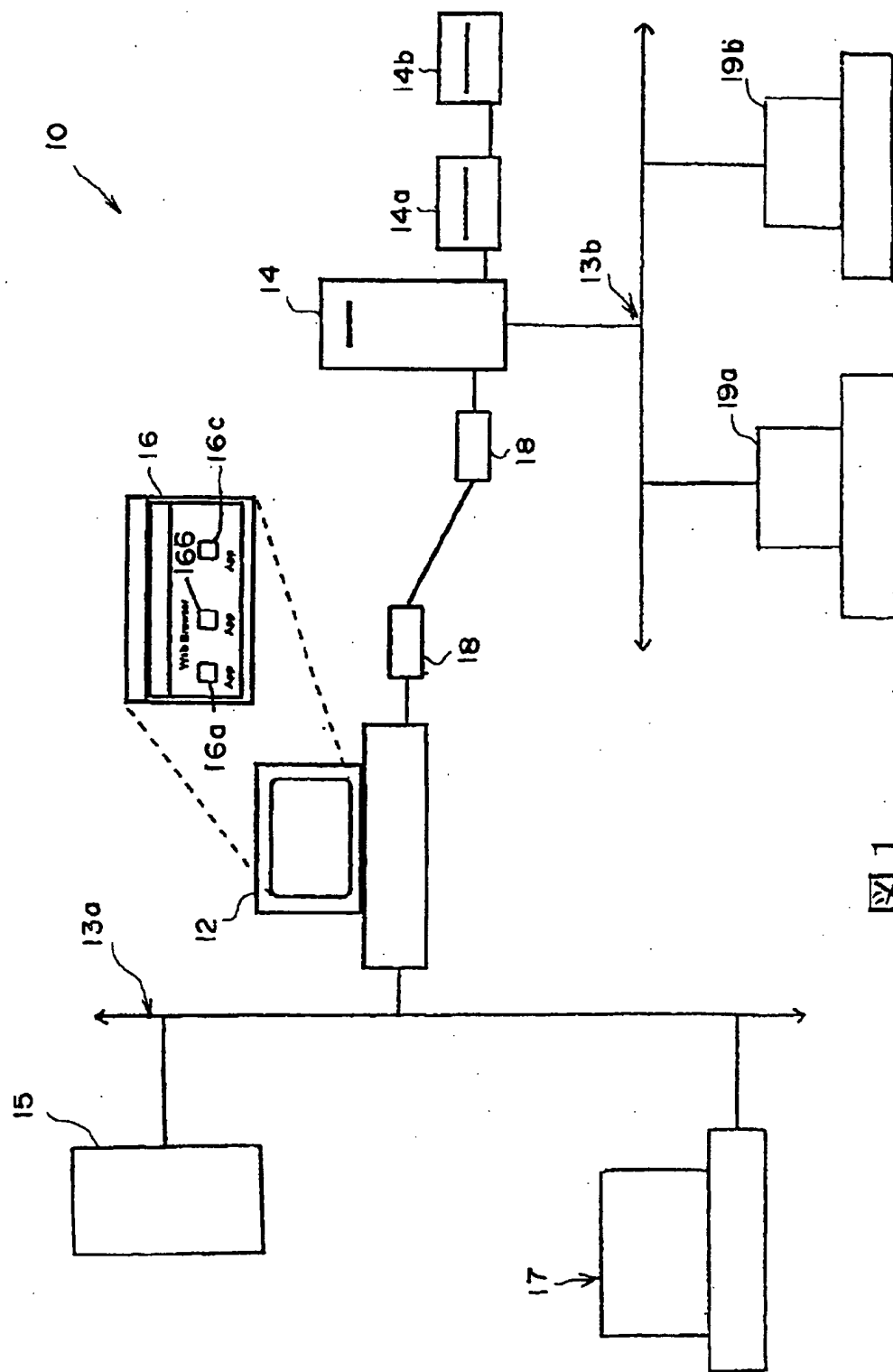


图 1

【図2】

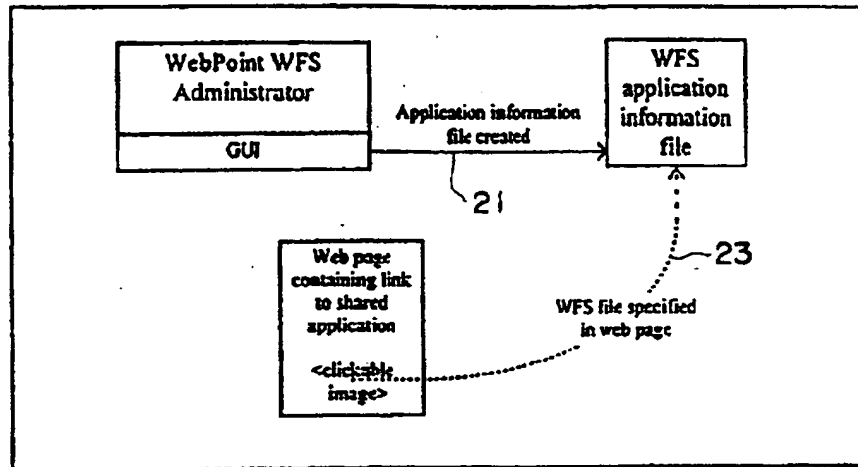
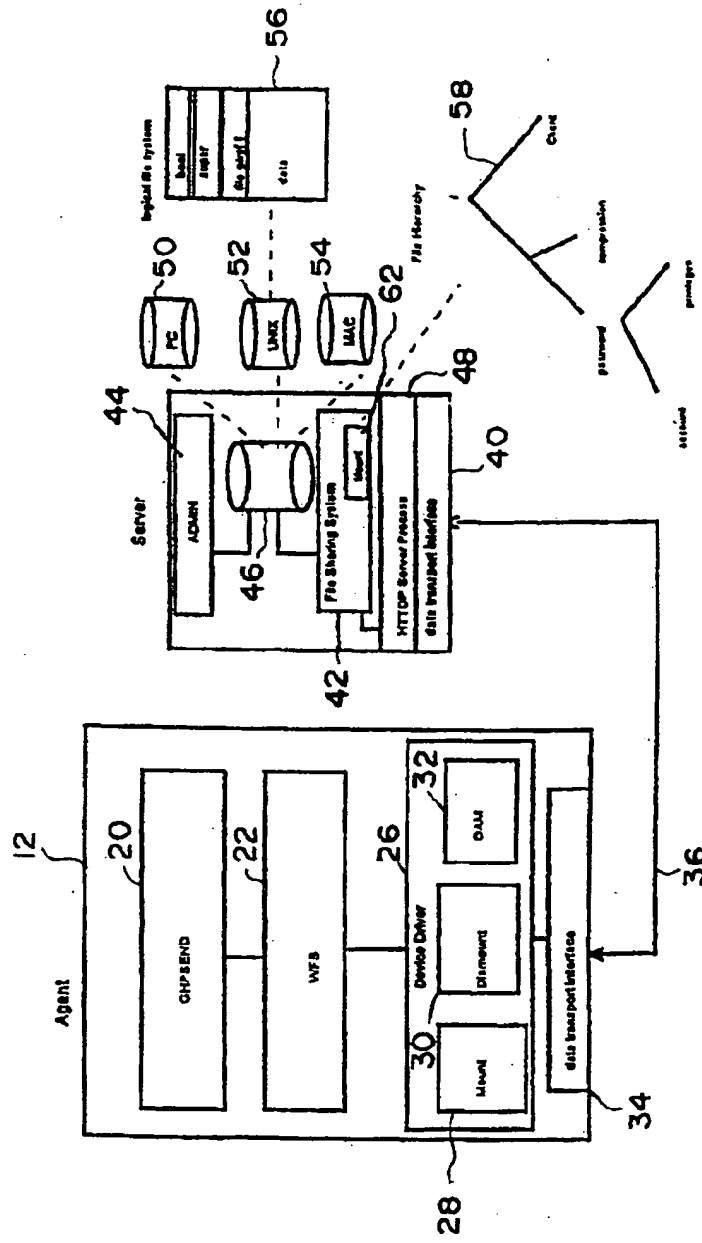


図 2

10



[3]

10

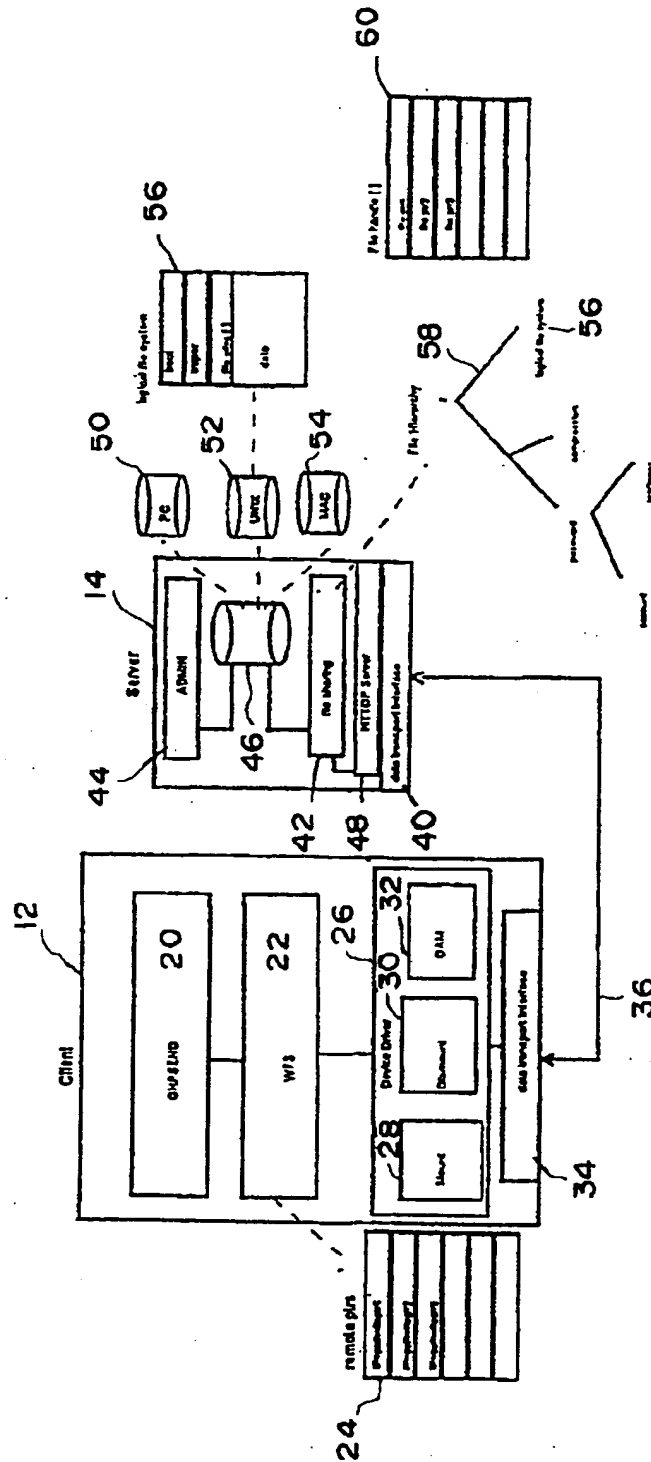
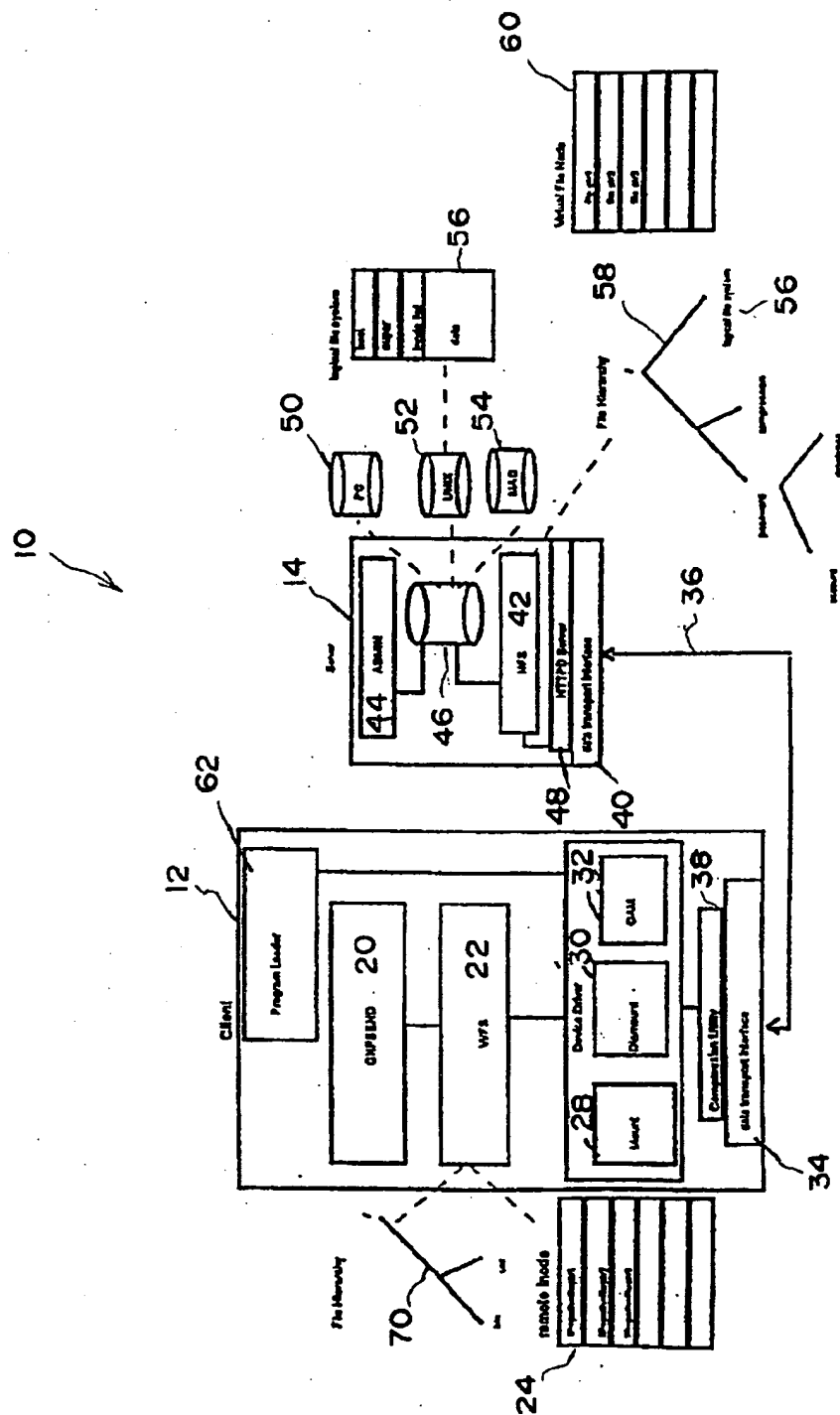


図4



𠂇

【図6】

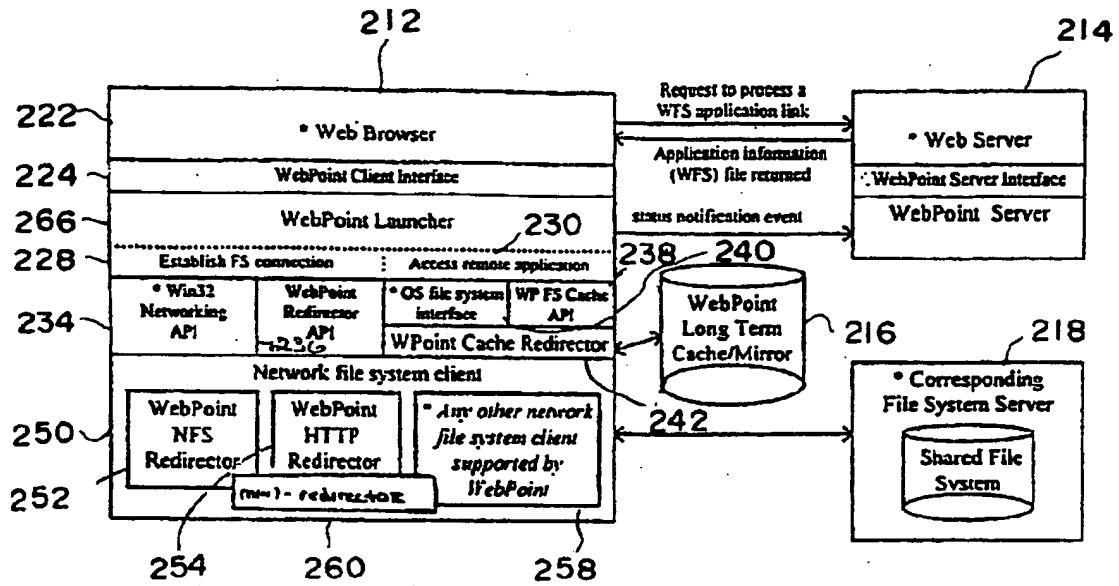


図 6

【図7】

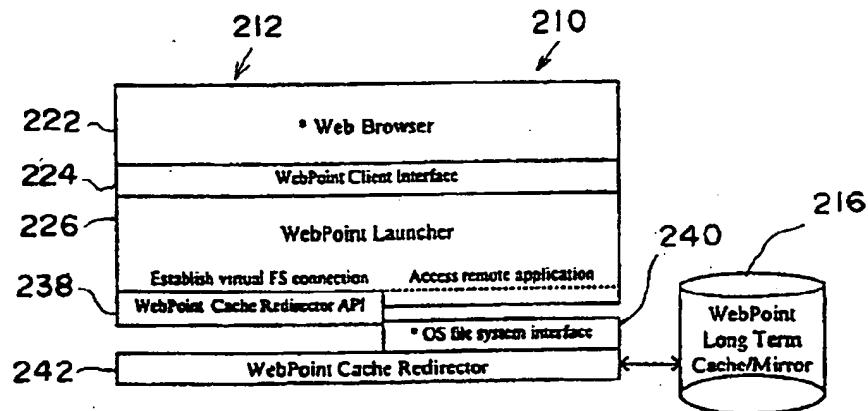


図 7

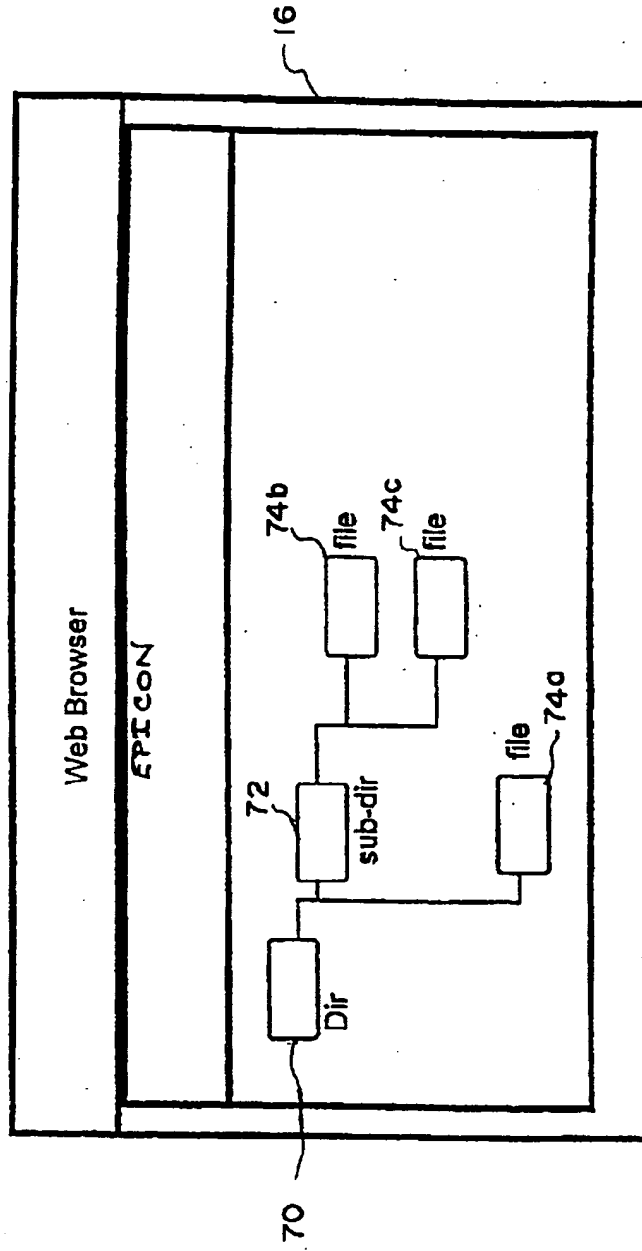


図8

【図 9】

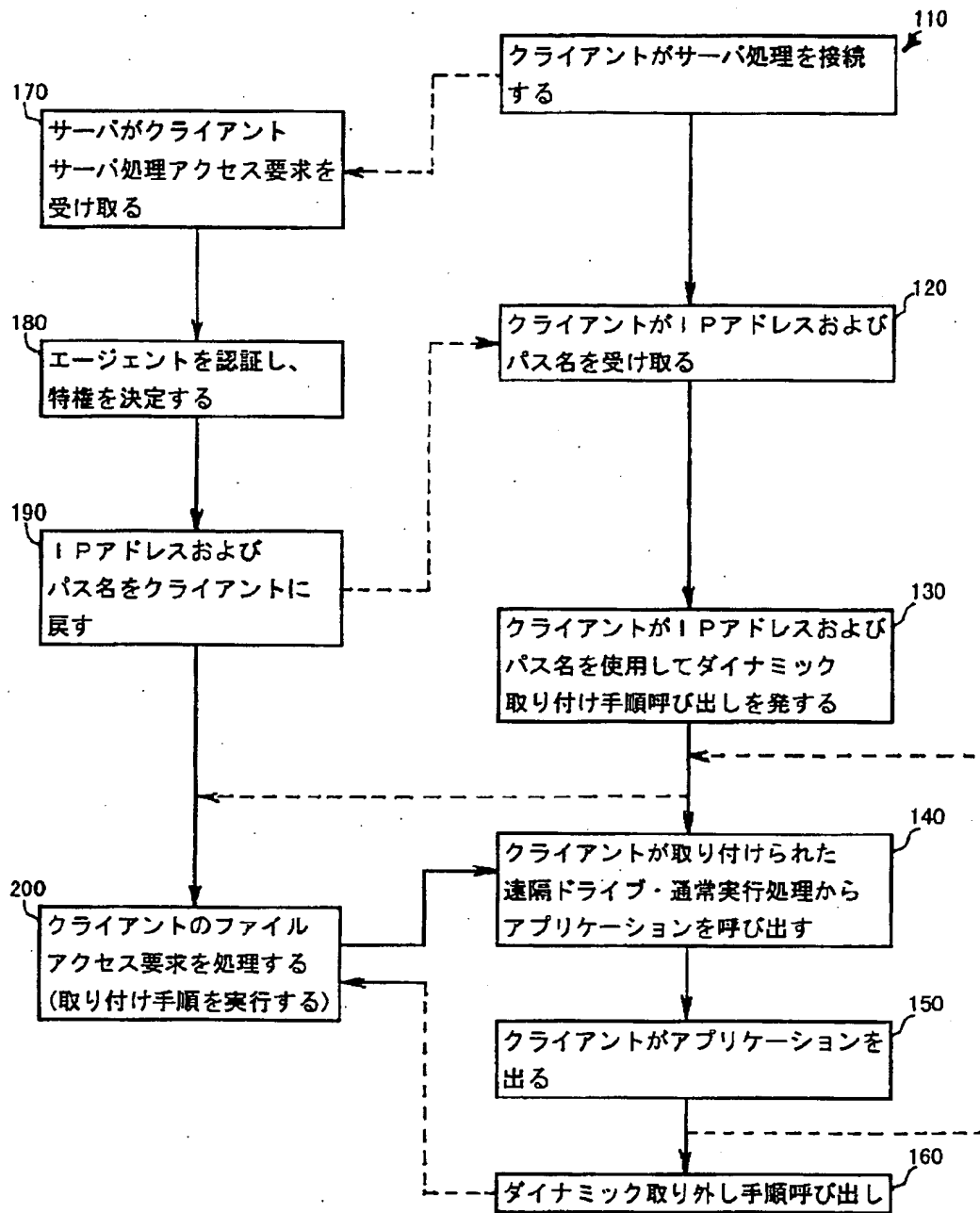


図 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/04194

A. CLASSIFICATION OF SUBJECT MATTER IPC(6) : G06F 9/445, 13/14, 17/30. US CL : 395/200.01, 200.03, 200.09, 200.12, 335, 610, 726-733, 762. According to International Patent Classification (IPC) or to both national classification and IPC				
B. FIELDS SEARCHED Minimum documentation searched (classification system followed by classification symbols) U.S. : G06F 9/445, 13/14, 17/30; 395/200.01, 200.03, 200.12, 335, 610, 726-733, 762. Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) APS Text and MAYA searches pending in parent application US 08/616,746.				
C. DOCUMENTS CONSIDERED TO BE RELEVANT				
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.		
A	US 5,280,627 A (FLAHERTY et al) 18 January 1994, please see abstract and Fig 1.	1, 19		
A	US 5,146,568 A (FLAHERTY et al) 08 September 1992, please see col. 4, lines 5-22.	1, 19		
A, P	US 5,504,892 A (ATSATT et al) 02 April 1996, please see the abstract and col. 9, lines 37-57.	1, 19		
A	US 5,423,034 A (COHEN-LEVY et al) 06 June 1995, abstract.	1, 19		
<input type="checkbox"/> Further documents are listed in the continuation of Box C. <input type="checkbox"/> See patent family annex.				
<table border="0"> <tr> <td> * Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document published on or after the international filing date "I" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reasons (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed </td> <td> "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "Z" document member of the same patent family </td> </tr> </table>			* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document published on or after the international filing date "I" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reasons (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "Z" document member of the same patent family
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier document published on or after the international filing date "I" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reasons (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "Z" document member of the same patent family			
Date of the actual completion of the international search 12 JUNE 1997		Date of mailing of the international search report 21 AUG 1997		
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230		Authorized officer E. Todd Vockz Telephone No. (703) 305-3900		

Form PCT/ISA/210 (second sheet) (July 1992)*

(81)指定国 EP(AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OA(BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), AP(GH, KE, LS, MW, SD, SZ, UG), EA(AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN

(72)発明者 ドメニコス, ジョージ シー

アメリカ合衆国 マサチューセッツ州

02150 チェルシー ナンバー203 ポート

スウェイン ウェイ 100